NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TEMPORALLY ADJUSTED COMPLEX AMBIGUITY FUNCTION MAPPING ALGORITHM FOR GEOLOCATING RADIO FREQUENCY SIGNALS

by

Andrew Moss

December 2014

Thesis Advisor:	Herschel H. Loomis
Thesis Co-Advisor:	Frank Kragh

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	**2. REPORT DATE** 12-19-2014	**3. REPORT TYPE AND DATES COVERED** Master's Thesis 06-18-2012 to 12-19-2014

4. TITLE AND SUBTITLE TEMPORALLY ADJUSTED COMPLEX AMBIGUITY FUNCTION MAPPING ALGORITHM FOR GEOLOCATING RADIO FREQUENCY SIGNALS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Andrew Moss	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Navy	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

The Complex Ambiguity Function (CAF) allows simultaneous estimates of the Time Difference of Arrival (TDOA) and Frequency Difference of Arrival (FDOA) for two received signals. The Complex Ambiguity Function Geo-Mapping (CAFMAP) algorithm then directly maps the CAF to geographic coordinates to provide a direct estimation of the emitter's position. The CAFMAP can only use short-duration CAFs, however, because the collector motion changes the system geometry over time. In an attempt to mitigate this shortfall and improve geolocation accuracy, the CAFMAP takes multiple CAF snapshots and sums the amplitudes of each. Unfortunately, this method does not provide the expected accuracy improvement, and a new method is sought.

This thesis reformulates the equations used in computing the CAF, in order to account for the collector's motion, and uses the results to derive a new CAFMAP algorithm. This new algorithm is implemented in MATLAB, and its results and characteristics analyzed. The conclusions are as follows: the new algorithm functions as intended, removes the accuracy limitations of the original method, and merits further investigation. Immediate future work should focus on ways to reduce its computation time and modifying the algorithm to account for 3-Dimensional reality, non-linear motion of collectors, and motion of the emitter.

14. SUBJECT TERMS RF Signals, Geolocation, CAF, CAFMAP, SIGINT	**15. NUMBER OF PAGES** 79
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

TEMPORALLY ADJUSTED COMPLEX AMBIGUITY FUNCTION MAPPING ALGORITHM FOR GEOLOCATING RADIO FREQUENCY SIGNALS

Andrew Moss
Lieutenant, United States Navy
B.S., Utah State University, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2014

Author: Andrew Moss

Approved by: Herschel H. Loomis
Thesis Advisor

Frank Kragh
Thesis Co-Advisor

R. Clark Robertson
Chair, Department of Electrical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Complex Ambiguity Function (CAF) allows simultaneous estimates of the Time Difference of Arrival (TDOA) and Frequency Difference of Arrival (FDOA) for two received signals. The Complex Ambiguity Function Geo-Mapping (CAFMAP) algorithm then directly maps the CAF to geographic coordinates to provide a direct estimation of the emitter's position. The CAFMAP can only use short-duration CAFs, however, because the collector motion changes the system geometry over time. In an attempt to mitigate this shortfall and improve geolocation accuracy, the CAFMAP takes multiple CAF snapshots and sums the amplitudes of each. Unfortunately, this method does not provide the expected accuracy improvement, and a new method is sought.

This thesis reformulates the equations used in computing the CAF, in order to account for the collector's motion, and uses the results to derive a new CAFMAP algorithm. This new algorithm is implemented in MATLAB, and its results and characteristics analyzed. The conclusions are as follows: the new algorithm functions as intended, removes the accuracy limitations of the original method, and merits further investigation. Immediate future work should focus on ways to reduce its computation time and modifying the algorithm to account for 3-Dimensional reality, non-linear motion of collectors, and motion of the emitter.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction 1

1.1 Background . . . 1

1.2 Objective . . . 3

1.3 Related Work . . . 3

1.4 Thesis Organization . . . 4

2 Overview of the CAF 5

2.1 Definition . . . 5

2.2 Simple Example . . . 6

2.3 CAF Performance . . . 11

3 A New Cross Ambiguity Function Geo-Mapping (CAFMAP) Algorithm 15

3.1 Developing the Basic Model . . . 17

3.2 Mapping Implementation . . . 22

3.3 Key Differences . . . 23

4 Results 25

4.1 Computational Complexity . . . 25

4.2 Moss Algorithm versus Hartwell Algorithm . . . 29

4.3 Conclusion of Results . . . 34

5 Future Work 35

5.1 Reduce Complexity . . . 35

5.2 Experimentally Determine Statistical Results . . . 37

5.3 Improve Geometry . . . 37

5.4 Conclusion . . . 37

Appendix: MATLAB Code 39

List of References 51

Initial Distribution List 53

List of Figures

Figure 1 Results of Hartwell's CAFMAP algorithm. xv

Figure 2 Hartwell CAFMAP results for varying signal lengths. xvii

Figure 3 Example geolocation geometry. xvii

Figure 4 CAFMAPS with dm = 100, SNR = 10 dB, and 30 snapshots. . . xx

Figure 5 Results of Hartwell's CAFMAP algorithm. 2

Figure 6 rect(t) Signal. 7

Figure 7 rect(t)rect$(t-1)$. 8

Figure 8 rect(t)rect$(t-(\Delta t-\tau))$. 10

Figure 9 Complex Ambiguity Function (CAF) of the example rect(t) signals, with offsets of 10 Hz in frequency, and 0.1 seconds in time. . . . 12

Figure 10 Results of Hartwell's CAFMAP algorithm. 15

Figure 11 Hartwell CAFMAP results for varying signal lengths. 18

Figure 12 Example geolocation geometry. 20

Figure 13 Moss algorithm with co-channel emitters at (10000,10000) and (6000,8000). 26

Figure 14 $N=2^{14}$, $dm=100$ m, full map. 28

Figure 15 $N=2^{22}$, $dm=100$ m, small map. 28

Figure 16 $N=2^{14}$, dm = 100, SNR = 10 dB, 10 Snapshots, Full Map. . . . 30

Figure 17 $N=2^{14}$, dm = 100, SNR = 10 dB, 30 Snapshots, Full Map. . . . 31

Figure 18 $N=2^{15}$, dm = 100, SNR = 30 dB, 10 Snapshots. 32

Figure 19 $N=2^{22}$, dm = 100, SNR = 30 dB, 10 Snapshots. 33

Figure 20 $N=2^{22}$, dm = 10, SNR = 30 dB, 10 Snapshots. 34

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Table 1 Number of operations performed. 27

Table 2 Test geometry parameters. 29

Table 3 Transmitted signal parameters. 29

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

3D	three-dimensional
BPSK	Binary Phase Shift Keying
CAF	Complex Ambiguity Function
CAFMAP	Cross Ambiguity Function Geo-Mapping
CI	Confidence Interval
DOD	Department of Defense
FDOA	Frequency Difference of Arrival
FFT	Fast Fourier Transform
FT	Fourier Transform
NPS	Naval Postgraduate School
NRL	Naval Research Laboratory
RF	Radio Frequency
SIGINT	Signals Intelligence
SNR	Signal-to-Noise Ratio
TDOA	Time Difference of Arrival

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Knowing the precise geographical position of adversaries and their equipment is critical to our ability to take action against them. Therefore, the accuracy of Department of Defense (DOD) Signals Intelligence (SIGINT) systems is of the utmost importance. These systems make use of a variety of methods to perform geolocation, each with their strengths and weaknesses [1], [2], [3]. This thesis is an attempt to improve upon the accuracy of one of these methods, called the Cross Ambiguity Function Geo-Mapping (CAFMAP) [4].

The CAFMAP method may be used when two spatially separate, moving collectors gather independent copies of the same emitted Radio Frequency (RF) signal. The Complex Ambiguity Function (CAF) of these two signals is then calculated, which generates a surface in the Time Difference of Arrival (TDOA) and Frequency Difference of Arrival (FDOA) domains. This surface peaks at the actual TDOA and FDOA values embedded in the signals by the physical geometry of the collector-emitter system. In the CAFMAP method, this surface is mapped to geographic coordinates and provides a three-dimensional (3D) map, peaking at the estimated location of the RF emitter. As shown in Figure 1, this method is effective, and is particularly good at locating multiple, co-channel emitters in close proximity [4].

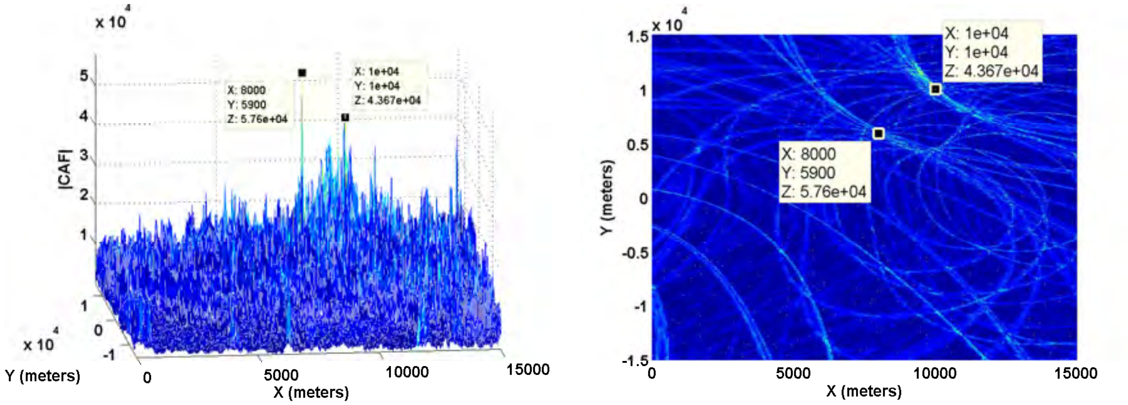


Figure 1. Results of Hartwell's CAFMAP algorithm.

The drawback to the CAFMAP method, however, is that it cannot be used with signals collected over long intervals. This is due to the changing collector-emitter geometry over time, which causes a smearing effect in the CAF. When mapped to geographic coordinates, this smearing manifests itself as noise in the form of spurious peaks in the map.

These spurius peaks increase the map's variance and confuse the geolocation estimate. An example of this is shown in Figure 2. As the signal lengths increase, the peaks become larger and the noise floor smaller, but when the signals become longer than $\sim 2^{20}$ samples, the noise becomes quite noticeable.

This work reformulates the CAF equations used in the CAFMAP algorithm, such that the collector-emitter geometry is updated with each successive sample of the collected signals. This allows for much longer signal collection and integration times without smearing, which, in turn, improves the geolocation accuracy of the CAFMAP method.

The CAF is mathematically defined as

$$A(\tau, f) = \int_0^T s_1(t) s_2^*(t+\tau) e^{-j2\pi f t} dt, \tag{1}$$

where signal collection starts at time $t = 0$, and ends at time $t = T$. Furthermore, when applied specifically to geolocating a received signal:

$$\begin{aligned} s_1(t) &= \text{analytic signal received at collector 1,} \\ s_2(t) &= \text{analytic signal received at collector 2,} \\ \tau &= \text{time difference, and} \\ f &= \text{frequency offset. [5]} \end{aligned}$$

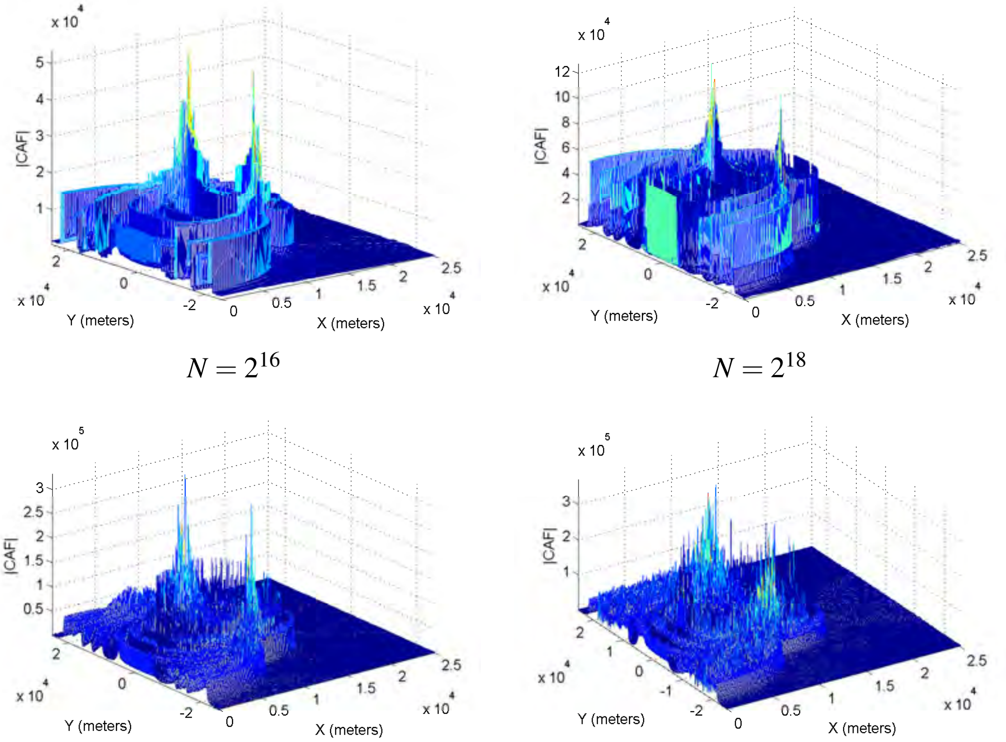
The magnitude of the CAF is then maximized when τ and f are equal to the TDOA and FDOA values inherent in the two received signals. These TDOA and FDOA values are defined as

$$\tau = \frac{|\vec{r}_2| - |\vec{r}_1|}{c} \text{ and,} \tag{2}$$

$$f = \frac{f_c}{c} \left[\frac{\vec{v_{c_2}} \cdot \vec{r}_2}{|\vec{r}_2|} - \frac{\vec{v_{c_1}} \cdot \vec{r}_1}{|\vec{r}_1|} \right]. \tag{3}$$

where f_c is the transmission carrier frequency, c is the speed of light, and the other factors are illustrated in Figure 3 [1], [2].

By allowing the $\vec{r}$ and $\vec{v}$ factors in (2) and (3) to be time varying, changing by the known collector parameters, and subsituting the result back into (1), we arrive at a new version of the CAF, that now accounts for the changing geometry over time. The result, when



$N = 2^{16}$ $N = 2^{18}$

$N = 2^{20}$ $N = 2^{21}$

Figure 2. Hartwell CAFMAP results for varying signal lengths.

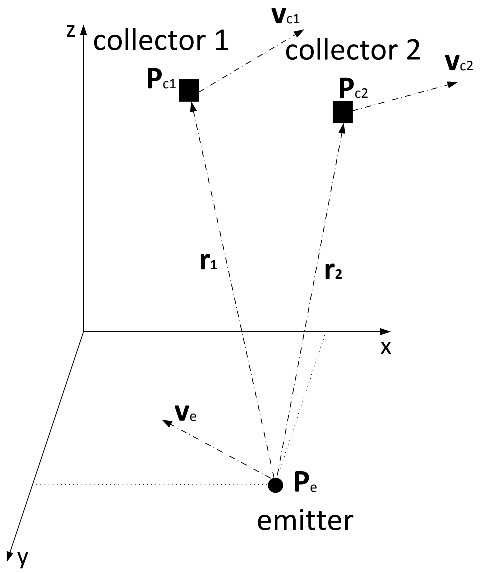


Figure 3. Example geolocation geometry.

converted to the digital domain for implementation, is shown in (4). The implementation of this modified CAF equation in the CAFMAP method is

$$A\left(x,y\right) = \sum_{n=n_0}^{n_0+f_sT-1} s_1\left[n\right] s_2^*\left[n + \text{ROUND}\left(\frac{|\vec{r_2}(nT_s)| - |\vec{r_1}(nT_s)|}{cT_s}\right)\right] e^{-j2\pi\frac{f_cn}{f_sc}\left[\frac{v\vec{c_2}\cdot\vec{r_2}(nT_s)}{|\vec{r_2}(nT_s)|} - \frac{v\vec{c_1}\cdot\vec{r_1}(nT_s)}{|\vec{r_1}(nT_s)|}\right]}, \quad (4)$$

and is herein referred to as the Moss algorithm. In (4), the factor $\left(\frac{|\vec{r_2}(t)|-|\vec{r_1}(t)|}{cT_s}\right)$ must be rounded to the nearest integer because it is in the digital domain, and corresponds to an offset in the sample index. As such, fractional offsets are not allowed as they would result in programming errors.

The results of the Moss algorithm, for a simple case, are shown in Figure 4. For comparison, the results of the original Hartwell algorithm, for an identical scenario, are shown alongside. It is seen that the Moss algorithm is still able to geolocate the emitter. As an additional benefit, it demonstrates a smoother, higher resolution map than the original Hartwell algorithm. Further test cases show that the Moss algorithm is still fully capable of detecting multiple co-channel emitters, and, although not conclusive, indicate that it does yield higher accuracy for longer signal collection times.

Statistically conclusive results were not achievable at this time due to time constraints and the high computational complexity involved in the current implementation of the Moss algorithm. Full maps, with long signal lengths, would take approximately 40 days to compute, and even the small and simple test cases conducted in this work required at least one day each to complete. This made it unfeasible to gather valid statistical results, or to fully investigate the long collection times that this algorithm was designed to use. So, although all data does indicate that improvement was achieved, and that the new algorithm merits further investigation, it does require more work before it can be made practically useful.

The key limitation of the Moss algorithm, as currently implemented, is the computation time required to find a result. Therefore, the first improvement necessary is to make the algorithm more efficient, such that the run-time is practical. With a reduction in run-time it will be possible to determine statistical results and to fully characterize the algorithm.

Some ideas on how to achieve this are:

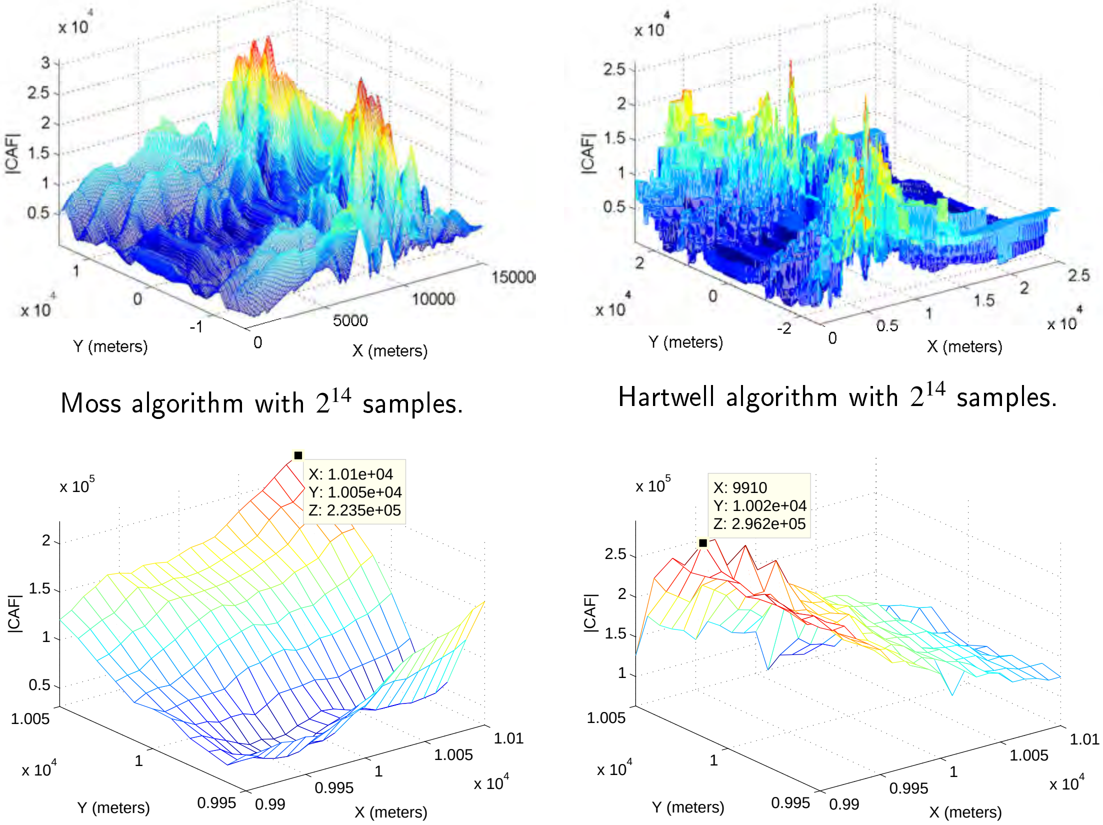
1. Port the algorithm to a more efficient language (i.e. C or Fortran).
2. Utilize parallel processing [6].
3. Implement an adaptive resolution regime.
4. Investigate block updates, vice updating geometry every sample.

Once this is accomplished, the reformulated equations should be re-examined. The equations formulated in this work assumed a very basic, linear geometry, and a stationary emitter. These assumptions were made for the sake of mathematical simplicity and clarity during proof of concept; however, they are generally not valid in real-world applications. Therefore, the model will need some modification to more truly reflect reality. Some key modifications required are:

1. Expand the model to three dimensions.
2. Change the equations of motion from linear to elliptical to account for orbital motion.
3. Allow for non-zero emitter velocities.

These modifications should not be difficult to accomplish, but will make the mathematical model significantly more complex.

In conclusion, it is shown that the reformulated equations for TDOA and FDOA are accurate, and can be used in a modified CAFMAP algorithm to account for the motion of the collectors over time. The resulting Moss algorithm is able to geolocate RF emitters with accuracy similar to that of the original Hartwell algorithm for shorter collection times, and apparently with greater accuracy for long collection times. The improvement comes, however, with the cost of high computational complexity. This is significant enough that it must be mitigated before the new algorithm can be practically implemented, but the work required is merited. Additionally, the approach used here, of modifying the TDOA and FDOA equations to account for collector motion, should be generally applicable to many other methods of geolocation that estimate TDOA and FDOA as a fundamental step. This could lead to much broader use of these results once the computational issues have been solved, and improve the geolocation accuracy of multiple methods in common use today.



Moss algorithm with 2^{14} samples.

Hartwell algorithm with 2^{14} samples.

Moss algorithm with 2^{22} samples.

Hartwell algorithm with 2^{22} samples.

Figure 4. CAFMAPS with dm = 100, SNR = 10 dB, and 30 snapshots.

List of References

[1] H. H. Loomis, “Geolocation of electromagnetic emitters,” Naval Postgraduate School, Monterey, CA, Tech. Rep. NPS-EC-00-003, October 2007.

[2] D. M. G. Price, “Mathematics of geolocation,” 2002, unpublished.

[3] P. D. Groves, *Principles of GNSS, Intertial, and Multisensor Integrated Navigation Systems*. Norwood, MA: Artech House, 2013.

[4] G. D. Hartwell, “Improved geo-spatial resolution using a modified approach to the complex ambiguity function (CAF),” master’s thesis, Dept. Elec. Eng., Naval Postgraduate School, September 2005.

[5] S. Stein, “Algorithms for ambiguity function processing,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 588–599, 1981.

[6] K. Singh *et al.*, “FFTW and complex ambiguity function performance on the maestro processor,” in *Aerospace Conference, 2011 IEEE*, March 2011, pp. 1–8.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would like to thank Dr. Herschel Loomis for the many hours he spent discussing research topics with me before I even decided on a direction. His knowledge and support throughout this process were invaluable, and I would not have been able to even broach the subject without his guidance. He continued to support me throughout my research and was always available to discuss my difficulties and findings.

I also owe a deep thanks to Dr. Frank Kragh. Without his patient assistance, I would likely still be trying to understand the mathematical complexities involved in CAF processing. He also providing many invaluable suggestions to help guide my work and writing, which has helped me to avoid many of the pitfalls I otherwise might have stumbled into.

Thanks also goes to Donna Miller, who provided me with the lab space and equipment that I needed to complete my work, and for our many conversations that helped me remain focused and motivated.

I would also like to express my gratitude to Dr. Charles Swenson. Although not directly involved with this work, he took the time to mentor and encourage me from the beginning of my academic career as an undergraduate student. He taught me to appreciate the complexities of big problems, and how they could be overcome to achieve great things.

Finally, no acknowledgement would be complete without an infinite thanks to my family; my parents, my wife and my children. They have always supported me and given me the time and resources that I needed to pursue my interests and desires, whatever they may be.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Background

In today's world of high-tech warfare, we have developed the ability to deploy virtually any type of ordnance quickly and accurately to any location on Earth. We can do this from a multitude of different platforms, which may be located safely many miles away and far beyond the reach of retaliation. These weapons have become so accurate that, with caution, they can be used in the midst of civilian populations to remove threats with minimal risk of collateral damage. What limits our ability to neutralize these threats is, quite often, insufficient knowledge of locations in order to strike.

Additionally, most conflicts today are asymmetric. We are often confronting adversaries without official, overt military forces and without the large-scale resources of a nation state. These covert groups substitute for their lack of resources by moving swiftly and secretly behind the mask of civilian populations and have repeatedly proven themselves capable of causing great harm to our nation and world peace. While our forces are more than capable of dealing with these people, they can only do so if they know who and where they are.

These facts indicate that one of the most crucial elements of modern warfare is precisely knowing the location of threats in real-time, or very close to it. Many tools must be used in collaboration to achieve this knowledge, one of which is Signals Intelligence (SIGINT). SIGINT methods involve collecting the Radio Frequency (RF) signals emitted by our adversary's own use of technology to gather information on their location and activities. Of particular importance to this thesis is the method of using satellites in orbit to collect emitted RF signals and to determine their location on the surface of the Earth.

There are many methods and algorithms for achieving this, most of which use a pair of satellites (collectors) to receive the signal and then compute a Time Difference of Arrival (TDOA) and Frequency Difference of Arrival (FDOA) between them [1], [2], [3]. These parameters can then be used to determine the location of the emitter. This is essentially the reverse of the process used by GPS receivers on the Earth to estimate one's position and velocity.

One of these methods was proposed by Al Buczek [4] of the Naval Research Laboratory (NRL) in the early 1990s and theoretically developed and tested in a 2005 master's thesis by Hartwell [5] at the Naval Postgraduate School (NPS). In this method, called the Cross Ambiguity Function Geo-Mapping (CAFMAP) method by Hartwell, the Complex Ambiguity Function (CAF) of the two signals received by a pair of collectors is calculated and then mapped to *xy* coordinates on the Earth's surface. The magnitude of the CAF forms a mathematical surface that peaks at the correct TDOA and FDOA values embedded in the signals. When this is mapped to the Earth's surface, the result is a function of position that peaks at the geographical coordinates of the emitter, as demonstrated in Figure 5.

The CAFMAP method was successfully demonstrated by Hartwell [5], but it suffered from a key weakness— its inability to use long duration signals. Attempts to do so resulted in a smearing of the CAF due to a change in the collector-emitter geometry over the collection time. This limits the best possible accuracy of Hartwell's method as the integration time is a fundamental factor in the precision of geolocating signals [6]. Hartwell was able to mitigate this deficiency, somewhat, by collecting multiple, short snapshots of the RF signals at spaced-out intervals, and then non-coherently summing the magnitudes of the resulting CAFs. This reduced the variance in the peaks of the resulting map and provided a better position estimate of the emitter, but it was still not an ideal solution. This thesis attempts to fundamentally correct this deficiency by modifying the CAF algorithm, such that arbitrarily long collection times are made possible without smearing the results. This correction should yield a version of the CAFMAP method where the estimation error can be made as small as desired by simply increasing the collection duration. The trade-off for

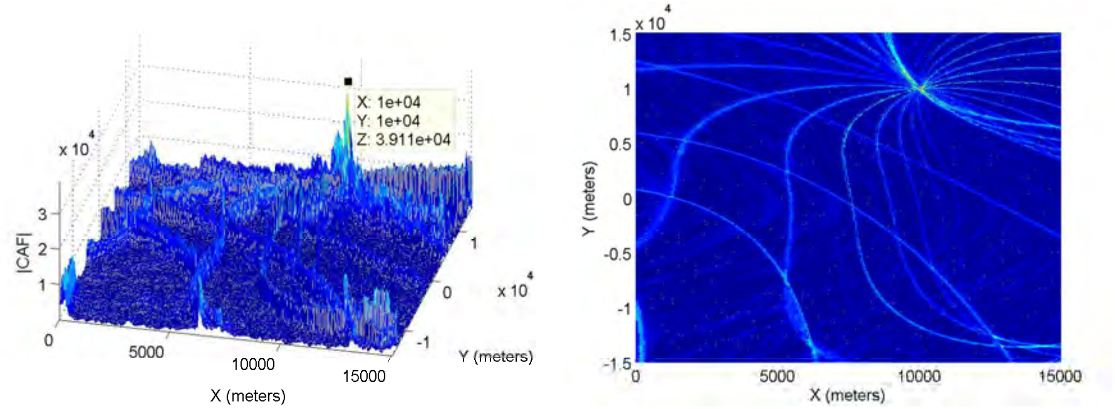


Figure 5. Results of Hartwell's CAFMAP algorithm.

this advantage being increased computational complexity and memory requirements.

1.2 Objective

The primary objective of this thesis is to derive a mathematical model of the CAF that accounts for the motion of the collectors and emitters in order to prevent the smearing of the CAF surface over long collection times. This will allow the CAFMAP method to be used with increased integration intervals, which, in turn will reduce the variance of the TDOA and FDOA peaks in the resulting map, thus providing a more accurate geolocation result. This work is focused on extending the capabilities of the CAFMAP, but the principles should generally apply to any geolocation technique utilizing the CAF to determine TDOA and FDOA values.

Upon successfully deriving a mathematical model, this study implements it in MATLAB to validate and characterize the results.

1.3 Related Work

Many papers, thesis and dissertations have been written on the topic of CAF processing and geolocating RF signals. The most directly relevant work to this thesis is that of Hartwell, where he developed and demonstrated the basic CAFMAP algorithm [5]. This thesis originated as an attempt to extend and improve Hartwell's work.

The fundamental concepts and algorithms employed in geolocation are discussed by Loomis [1], Price [2], and Groves [3]. These fundamentals have been in practical use for several decades and still form the foundations of SIGINT geolocation. The CAF is also a critical concept in this thesis and its definition, uses and methods of computation are developed in [6], [7], [8], and [9]. Skolnik also provides some useful insight into the uses of the CAF in RADAR [10], where the function found its original use.

Johnson's Naval Postgraduate School thesis [11] is also of great value. Johnson created and tested the signal generating functions used in this work and provided a practical implementation of using the CAF to perform geolocation.

1.4 Thesis Organization

This thesis is organized into five chapters. Chapter 2 gives an overview of the CAF and a demonstration of how it can be used to estimate TDOA and FDOA parameters simultaneously. It also introduces the basic factors affecting the accuracy of the CAF. Chapter 3 develops the mathematical model for the new CAFMAP algorithm and discusses its implementation in MATLAB. Chapter 4 presents results and compares them to those of Hartwell's original method. Chapter 5 concludes this thesis by presenting some ideas for future work and ways that could make this new algorithm more useful in a practical application.

CHAPTER 2:
Overview of the CAF

This study is focused on extending the capabilities of the CAF. Thus, the first step is to arrive at a good understanding of what a CAF is and how it is used in the field of geolocation. In essence the CAF is a correlation function that takes a time and frequency offset as its independent variables and peaks when the values of those variables match the inherent offsets embedded in a received signal. It has found widespread use in the fields of radar detection and geolocation because of its ability to simultaneously estimate both TDOA and FDOA values [1], [6], [7], [9], [10], [11].

2.1 Definition

There are multiple definitions of the CAF but the most common, and the one that will be used in this text, is that given by Stein [6] as

$$A(\tau, f) = \int_0^T s_1(t) s_2^*(t+\tau) e^{-j2\pi f t} dt. \tag{2.1}$$

The integration limit, T, is the length of time over which the signals are collected. Furthermore, when applied specifically to geolocating a received signal:

$s_1(t)$ = analytic signal received at collector 1,

$s_2(t)$ = analytic signal received at collector 2,

τ = time difference, and

f = frequency offset.

2.2 Simple Example

To understand how this works we will use a simple signal, $\text{rect}(t)$, as an example. This signal is shown in Figure 6 and is defined as:

$$\text{rect}(t) = \begin{cases} 1, & \text{if } -\frac{1}{2} \leq t \leq \frac{1}{2} \\ 0, & \text{Otherwise.} \end{cases} \tag{2.2}$$

In a geolocation problem, a signal is emitted by a transmitter. This signal is then detected by two separate collectors with different positions and/or velocities. These received signals are $s_1(t)$ and $s_2(t)$. Both received signals are the original $\text{rect}(t)$ signal that was transmitted, but they will have slightly different times of arrival and frequencies in addition to some added noise. The times of arrival are different because the two collectors are in different positions, therefore the path length from the emitter to each collector is different. The difference in path length causes a difference in time of arrival. Similarly, the collectors will typically have different velocity vectors. The relative velocity between each collector and the emitter will cause a Doppler shift, but since the relative velocity vectors are not equal, the amount of Doppler shift will not be either. This leads to a difference in frequency between the two received signals. The following example shows how these offsets are determined using the CAF. For this example, the following terms will be used:

$$\begin{aligned} \tau &= \text{possible time difference parameters,} \\ f &= \text{possible frequency offset parameters,} \\ \Delta t &= \text{actual time difference between signals, and} \\ \Delta f &= \text{actual frequency difference between signals.} \end{aligned}$$

The objective is to calculate the CAF, which is defined in (2.1). This definition follows that given by Stein [6], and is used in this work, but the function can also be described in a more general form, with infinite bounds of integration, such as the definition found in Ramirez et al. [9]. The two definitions are equivalent, in practice, as the collected signals are of finite length, starting at $t = 0$ when the collection starts, and ending at $t = T$ when the

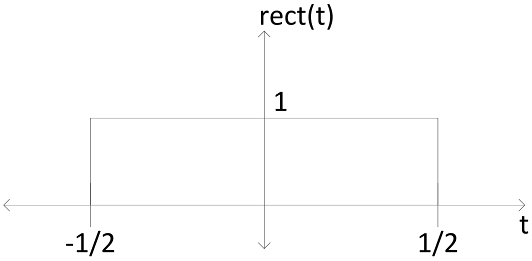


Figure 6. rect(t) Signal.

collection ends. In practical applications, the Stein definition makes more intuitive sense, and is the definition we use; but in this example, we will revert to the infinite bounds for mathematical convenience. Understand that they are equivalent in practice. Thus, we let

$$A(\tau, f) = \int_{-\infty}^{\infty} s_1(t) s_2^*(t+\tau) e^{-j2\pi f t} dt, \tag{2.3}$$

where

$$s_1(t) = \text{rect}(t) e^{j2\pi f_c t} \text{ and} \tag{2.4}$$

$$s_2(t) = \text{rect}(t - \Delta t) e^{j2\pi (f_c - \Delta f) t}. \tag{2.5}$$

For simplicity, the problem will be broken down into segments. Let,

$$m(t) = s_1(t) s_2^*(t+\tau), \tag{2.6}$$

which is known as the mixing function, and yields

$$\begin{aligned} m(t) &= s_1(t) s_2^*(t+\tau) && (2.7) \\ &= \text{rect}(t) e^{j2\pi f_c t} \text{rect}(t - \Delta t + \tau) e^{-j2\pi (f_c - \Delta f)(t+\tau)} && (2.8) \\ &= \text{rect}(t) \text{rect}(t - \Delta t + \tau) e^{-j2\pi (f_c \tau - \Delta f t - \Delta f \tau)}. && (2.9) \end{aligned}$$

The mixing function is then combined with the exponential term of the CAF, taking the frequency offset into account, yielding

$$m(t)e^{-j2\pi ft} = \text{rect}(t)\text{rect}(t-\Delta t+\tau)e^{-j2\pi(f_c\tau-\Delta ft-\Delta f\tau)}e^{-j2\pi ft} \tag{2.10}$$
$$= \text{rect}(t)\text{rect}(t-\Delta t+\tau)e^{-j2\pi(f_c\tau-\Delta ft-\Delta f\tau+ft)}. \tag{2.11}$$

The integral is then calculated yielding

$$A\left(\tau,f\right) = \int_{-\infty}^{\infty}\text{rect}(t)\text{rect}(t-\Delta t+\tau)e^{-j2\pi(f_c\tau-\Delta ft-\Delta f\tau+ft)}dt \tag{2.12}$$
$$= e^{-j2\pi(f_c-\Delta f)\tau}\int_{-\infty}^{\infty}\text{rect}(t)\text{rect}(t-(\Delta t-\tau))e^{-j2\pi(f-\Delta f)t}dt. \tag{2.13}$$

In order to develop an understanding of the CAF it is useful to look at the parts of this integral, rather than just finding the analytic solution. To see the results of the above integration we first neglect the frequency shift and just examine the Fourier transform of the two rect() functions.

If $|\Delta t-\tau| \geq 1$ then,

$$\text{rect}(t)\text{rect}(t-(\Delta t-\tau)) = 0 \tag{2.14}$$
$$\rightarrow A(\tau,f) = 0. \tag{2.15}$$

This is illustrated in Figure 7. If $|\Delta t-\tau| \geq 1$ then the two rect() functions have zero overlap and their product is zero, making the CAF equal zero.

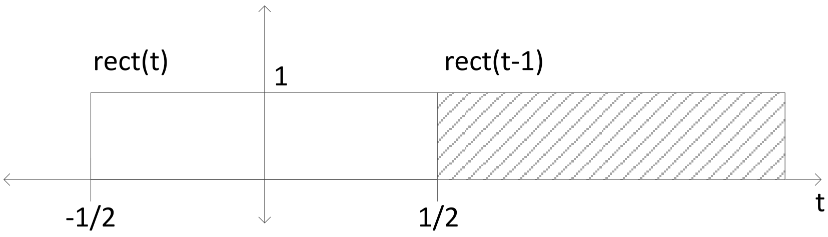


Figure 7. $\text{rect}(t)\text{rect}(t-1)$.

Otherwise, if $|\Delta t - \tau| = 0$ then,

$$\begin{aligned} \mathrm{rect}(t)\mathrm{rect}\left((t - (\Delta t - \tau)\right) &= \mathrm{rect}^2(t) \quad (2.16)\\ &= \mathrm{rect}(t), \quad (2.17) \end{aligned}$$

and since

$$\int_{\infty}^{\infty} \mathrm{rect}(t) e^{-j2\pi(f-\Delta f)t} dt = \mathrm{sinc}(f - \Delta f) \tag{2.18}$$

$$A(\tau = \Delta t, f) = e^{-j2\pi(f_c - \Delta f)\tau} \mathrm{sinc}(f - \Delta f). \tag{2.19}$$

The first factor is a complex phasor, meaning that it will change the phase of the result, but the magnitude will remain unity. The second factor is the well-known sinc() function, and its magnitude will equal one if $|f - \Delta f| = 0$, and will oscillate with decreasing magnitude otherwise. From this, it is seen that $|A(\tau, f)|$ will always be maximized when both $\tau = \Delta t$ and $f = \Delta f$.

The final case is where $0 < |\Delta t - \tau| < 1$. As shown in Figure 8, the mixing function $m(t) = \mathrm{rect}(t)\mathrm{rect}\left(t - (\Delta t - \tau)\right)$ becomes the shifted and scaled $\mathrm{rect}(t)$ function

$$m(t) = \mathrm{rect}\left(\frac{t - \frac{\Delta t - \tau}{2}}{1 - |\Delta t - \tau|}\right). \tag{2.20}$$

This gives a CAF equation of

$$A\left(0 < |\Delta t - \tau| < 1, f\right) = e^{-j2\pi(f_c - \Delta f)\tau} \int_{\infty}^{\infty} \mathrm{rect}\left(\frac{t - \frac{\Delta t - \tau}{2}}{1 - |\Delta t - \tau|}\right) e^{-j2\pi(f-\Delta f)t} dt. \tag{2.21}$$

It can be seen that the integral in (2.21) is just the Fourier Transform (FT) of (2.20), from which we can make use of the FT shift and scaling properties. Doing so yields

$$\mathrm{rect}\left(\frac{t - \frac{\Delta t - \tau}{2}}{1 - |\Delta t - \tau|}\right) \Leftrightarrow e^{-j2\pi\left(\frac{\Delta t - \tau}{2}\right)(f - \Delta f)} |1 - |\Delta t - \tau|| \mathrm{sinc}\left\{\left[1 - |\Delta t - \tau|\right](f - \Delta f)\right\} \tag{2.22}$$

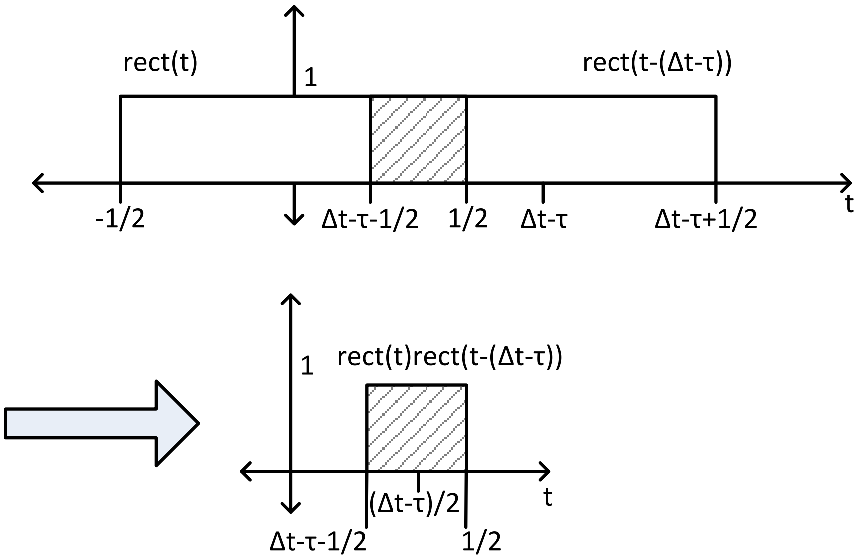


Figure 8. $\text{rect}(t)\text{rect}(t-(\Delta t-\tau))$.

$$\Rightarrow A\left(0<|\Delta t-\tau|<1,f\right)= e^{-j2\pi(f_c-\Delta f)\tau}e^{-j2\pi\left(\frac{\Delta t-\tau}{2}\right)(f-\Delta f)}|1-|\Delta t-\tau||\text{sinc}\left\{\left[1-|\Delta t-\tau|\right](f-\Delta f)\right\}. \quad (2.23)$$

Importantly, the boundaries of (2.23) agree with the previous cases. When $|\Delta t-\tau|=0$, then the term $[1-|\Delta t-\tau|]$ equals 1, and the sinc term becomes $\text{sinc}(f-\Delta f)$. This is maximized when $|f-\Delta f|=0$. When $|\Delta t-\tau|=1$, then $[1-|\Delta t-\tau|]=0$ and forces the entire function to zero. In between these boundaries, it can be seen that as $|\Delta t-\tau|$ increases, the coefficient term $[1-|\Delta t-\tau|]$ becomes smaller, which reduces the magnitude of the CAF. Therefore, the CAF will have a higher magnitude when the difference between τ and Δt is smallest. At the same time, the sinc() function is maximized when the difference between f and Δf is smallest and decreases as the difference is increased. Additionally, both exponential factors will cause oscillations, but their magnitude is 1 when $|\Delta t-\tau|=0$ and $|f-\Delta f|=0$ and can never be greater for any other values.

So it can be seen in this example that when the CAF surface is plotted with respect to τ and f, as shown in Figure 9, there will be a peak at the location corresponding to the actual signal offsets, Δt and Δf. This demonstrates that by numerically calculating the CAF of a signal, received by two different collectors, we can arrive at a good estimate of the values for both the TDOA and the FDOA. There will, however, be noise is this estimate, mostly due to noise inherent in the signals themselves. Therefore, one of our goals when utilizing the CAF for geolocation is to minimize the variance in the TDOA and FDOA estimates, in order to maximize accuracy. Of note in Figure 9 is that the variance in the FDOA is practically zero. This is due to the infinite bandwidth of the rect(t) signals used in this example and this relationship will be further discussed in the next section.

2.3 CAF Performance

Along with his work on how to compute CAF surfaces, Stein also derived the theoretical accuracies of both TDOA and FDOA estimates. These are

$$\sigma_{TDOA} = \frac{1}{\beta}\frac{1}{\sqrt{BT\gamma}} \tag{2.24}$$

$$\sigma_{FDOA} = \frac{1}{T_e}\frac{1}{\sqrt{BT\gamma}}, \tag{2.25}$$

with the terms defined as:

$T =$ signal collection time, i.e. length of $s_1(t)$ and $s_2(t)$,

$B =$ noise bandwidth at receiver input, assumed the same for both collectors,

$\beta =$ RMS bandwidth of the received signal,

$T_e =$ RMS integration time, and

$\gamma =$ effective input SNR.

Furthermore,

$$\beta = 2\pi\left[\frac{\int_{-\infty}^{\infty} f^2 W_s(f)df}{\int_{-\infty}^{\infty} W_s(f)df}\right]^{\frac{1}{2}} \tag{2.26}$$

where $W_s(f) =$ signal power density spectrum,

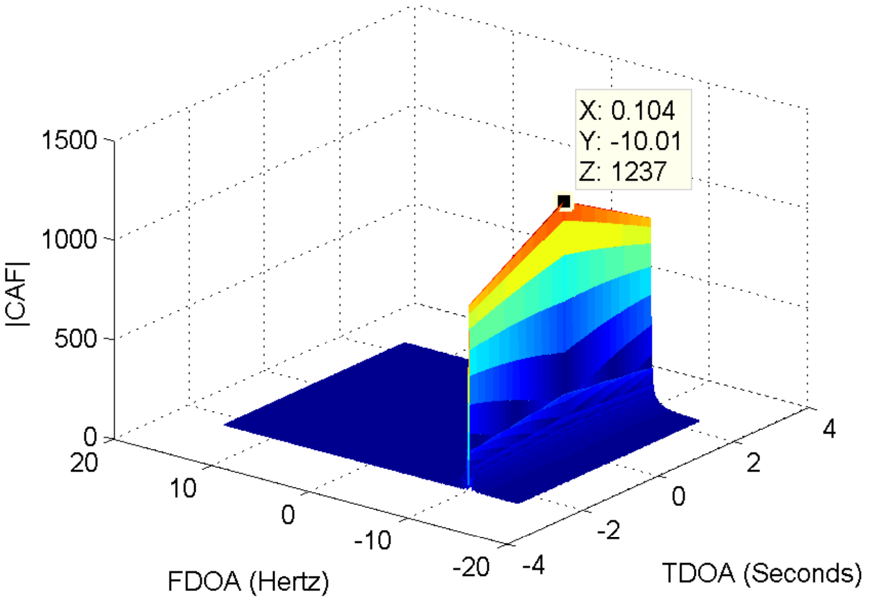


Figure 9. CAF of the example rect(t) signals, with offsets of 10 Hz in frequency, and 0.1 seconds in time.

$$T_e = 2\pi \left[\frac{\int_{-\infty}^{\infty} t^2 |u(t)|^2 dt}{\int_{-\infty}^{\infty} |u(t)|^2 dt} \right]^{\frac{1}{2}}, \tag{2.27}$$

where $u(t)$ is the waveform that was originally transmitted, and

$$\frac{1}{\gamma} = \frac{1}{2} \left[\frac{1}{\gamma_1} + \frac{1}{\gamma_2} + \frac{1}{\gamma_1 \gamma_2} \right], \tag{2.28}$$

where γ_1 and γ_2 are the signal to noise ratio's in the signal received by each collector [6].

It is clear that in order to minimize the variance in both TDOA and FDOA dimensions, we need to maximize the five terms in (2.24) and (2.25). B, β and γ are all dependent on the signal that is transmitted and, therefore, out of the collectors' control. That leaves T and T_e, which are both measures of the signal collection time. Thus, if we want to reduce the variance in the CAF peaks, and thereby improve the geolocation estimate, it is necessary to increase the signal collection time as much as possible.

This chapter has developed an understanding of the CAF and the key factors affecting its accuracy. The next chapter will introduce Hartwell's CAFMAP method of geolocation, and explain its inherent limitations regarding signal collection time. It will then explore how we might reformulate the CAF, such that the inherent motion of the collectors is accounted for. In principle, this will allow us to collect signals for as long as necessary to achieve our desired level of accuracy. In reality there will be other limiting factors, such as latency, memory, and processing capability, but the physical limitations of the CAFMAP method will have been overcome.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3: A New CAFMAP Algorithm

Hartwell demonstrated a novel method of using the CAF for geolocation in his master's thesis which he called the CAFMAP [5]. Traditional methods use the CAF to determine the TDOA and FDOA values, and then utilize a follow-on method, such as Newton-Raphson, to invert the TDOA and FDOA equations in order to arrive at geographic coordinates [1], [2]. The CAFMAP method, instead, maps the CAF directly to geographic coordinates. This results in a surface, or image, that peaks at the most likely location of the RF emitter.

Hartwell's approach for doing this was to create a grid map of *x* and *y* geographic coordinates and calculate what the expected TDOA and FDOA values would be if the emitter were located in each of those grid positions. He then calculated the actual CAF surface from the collected signals and, by cross-referencing the TDOA and FDOA values of this surface and the grid points, was able to map the CAF into *x* and *y* coordinates. The result was the CAF surface on an *xy*-coordinate, geographic plane. Hartwell's results were quite promising, particularly when dealing with multiple co-channel emitters as shown in Figure 10.

Unfortunately, it also relied upon constant signal parameters to be able to perform the cross-referencing and also to be able to calculate the CAF efficiently using Fast Fourier Transform (FFT) techniques. The implications of this restriction are shown in Figure 11. As the signal lengths are increased the CAF peaks become greater in magnitude and drop off more sharply. This means that the geolocation accuracy will be improved, as expected,

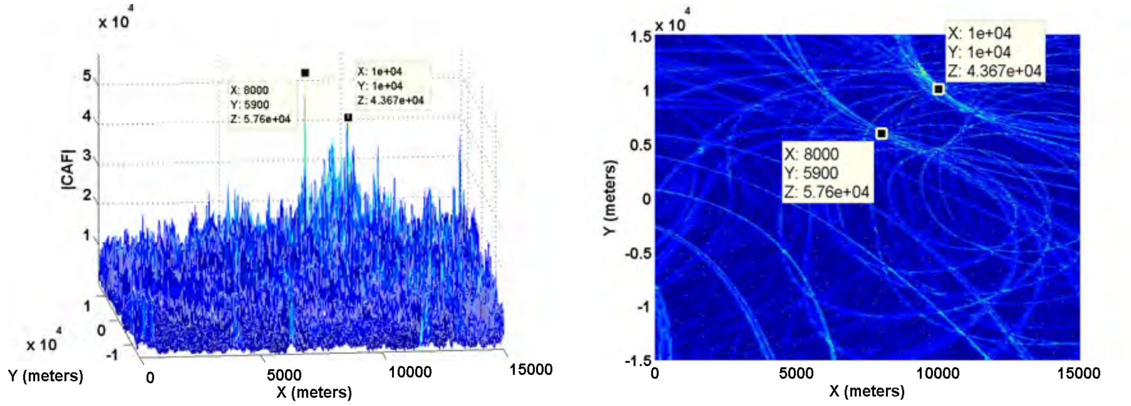


Figure 10. Results of Hartwell's CAFMAP algorithm.

for longer collection and integration times. However, when the signal lengths reach approximately 2^{20} samples long we can begin to see more noise in the form of spurious peaks away from the true emitter location. A signal length of 2^{20} samples corresponds to a collection time of approximately 10.5 seconds for the typical sampling rate of 100 kHz used in this example. This noise arises due to the fact that the collector-emitter geometry has changed over the length of the signal collection interval, and it results in geolocation accuracy becoming worse with longer collection times, rather than better.

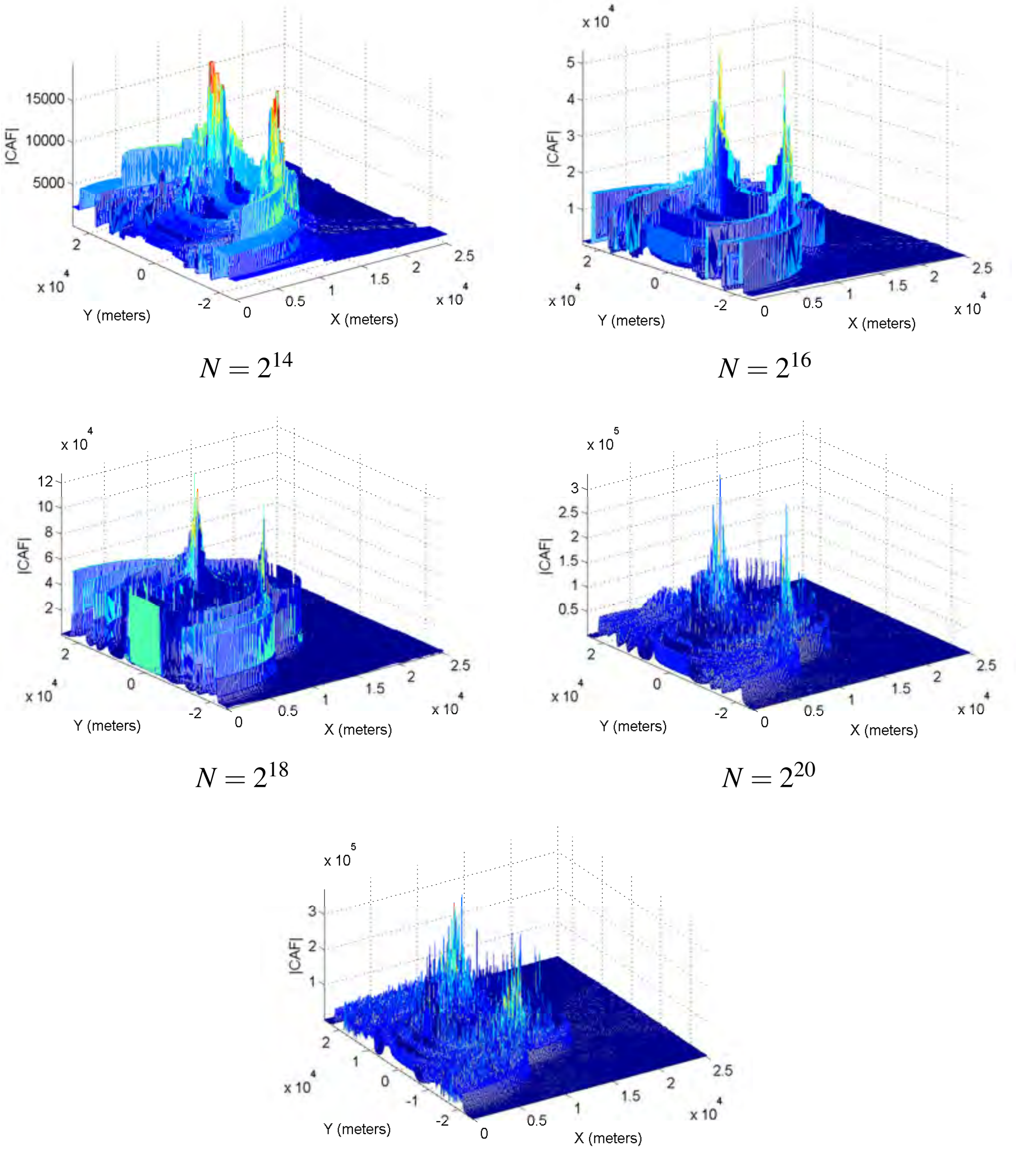
Hartwell recognized this limitation and made an effort to mitigate it. He did this by breaking the signal collection up into short duration snapshots, with a time gap between each. He then calculated the CAFMAP for each of these snapshots and non-coherently summed their magnitudes. The idea of this being that each snapshot would have its own noise, but the same peaks where emitters were located. Thus, by summing the snapshot magnitudes, the surface energy at the true emitter locations would keep increasing at a greater rate than the side lobes. This mitigation technique was partially successful. It allowed for longer effective signal collections, and yielded correspondingly narrower peaks, but it also tended to create its own noise. Much of this is due to the non-coherent summing used by Hartwell. With no phase information there is no destructive interference, so the noise floor is always increasing. Eventually this makes it difficult to distinguish peaks from the noise. Additionally there is a limit to how narrow the peaks can be made with this technique. Since each CAF snapshot must be from a short duration signal, there will be a relatively greater variance in the peaks being summed. Summing them will cause the center of the peak to rise faster than the surrounding surface, but it will not remove the energy of the individual wide peaks [5].

The goal of this thesis is to completely remove this physical limitation. This is accomplished by modifying the CAF equations used in the CAFMAP method, such that system geometry is updated throughout the signal collection period. In principle, this allows us to collect signals for as long as necessary to achieve our desired level of accuracy. In reality there will be other limiting factors, such as latency, memory, and processing capability, but the physical limitations of the CAFMAP method will have been overcome.

Initially, a variety of modifications to Hartwell's established algorithm were attempted to achieve the desired result. After some effort, however, it was realized that the basic architecture of Hartwell's algorithm was incompatible with the objective. Therefore, the basic formulation of the problem was revisited as follows.

3.1 Developing the Basic Model

The critical aspect of the problem is the geometry between the collectors and the emitter. A generic example geometry is shown in Figure 12, which will be used to derive a new model. A three-dimensional (3D) geometry is used for generality, however, certain restrictions will be enacted during this development in order to maintain mathematical simplicity. The standard CAF is a function of TDOA and FDOA, but our objective requires that it be made into a function of position and time, or, x, y, z and t. Accomplishing this is the first step in developing the new model.



$N = 2^{14}$

$N = 2^{16}$

$N = 2^{18}$

$N = 2^{20}$

$N = 2^{21}$

Figure 11. Hartwell CAFMAP results for varying signal lengths.

3.1.1 TDOA

For simplicity, TDOA and FDOA will be transformed separately, starting with TDOA. TDOA is simply the difference in time that is required for the signal to travel from the emitter to the two collectors. That is,

$$\tau = \frac{|\vec{r}_2| - |\vec{r}_1|}{c} \tag{3.1}$$

where $c =$ speed of light $\approx 2.998 \times 10^8 m/s$ [1], [2].

Or, in the terms of Figure 12 and using the velocity of the collectors and emitter, which are allowed to be time-variant,

$$\begin{aligned}
\tau(t) &= \frac{\sqrt{\left|\vec{P}_e(t) - \vec{P}_{c_2}(t)\right|^2} - \sqrt{\left|\vec{P}_e(t) - \vec{P}_{c_1}(t)\right|^2}}{c} && (3.2)\\
&= \frac{\sqrt{\left|\left(\vec{P}_e + \vec{v}_e \cdot t\right) - \left(\vec{P}_{c_2} + \vec{v}_{c_2} \cdot t\right)\right|^2} - \sqrt{\left|\left(\vec{P}_e + \vec{v}_e \cdot t\right) - \left(\vec{P}_{c_1} + \vec{v}_{c_1} \cdot t\right)\right|^2}}{c}. && (3.3)
\end{aligned}$$

For the remainder of this work it is assumed that $\vec{v}_e = 0$, i.e. only stationary emitters are considered. Examining the effects of this assumption will be left as a future work. With this assumption (3.2) simplifies back to

$$\tau(t) = \frac{|\vec{r}_2(t)| - |\vec{r}_1(t)|}{c}, \tag{3.4}$$

where

$$\vec{r}_1(t) = \vec{P}_e - \left(\vec{P}_{c_1} + \vec{v}_{c_1} \cdot t\right) \text{ and} \tag{3.5}$$

$$\vec{r}_2(t) = \vec{P}_e - \left(\vec{P}_{c_2} + \vec{v}_{c_1} \cdot t\right). \tag{3.6}$$

This successfully converts the TDOA into a function of position and time. The velocities of the collectors must be known, but in general they are and this is not a problem.

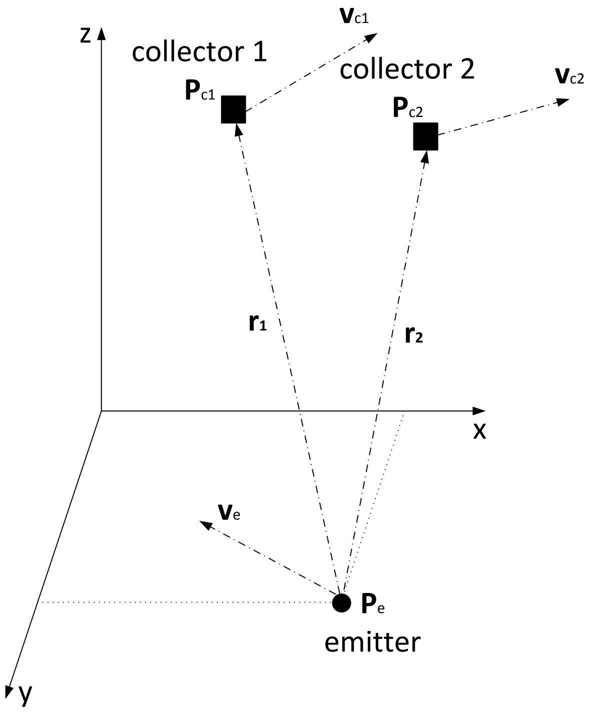


Figure 12. Example geolocation geometry.

3.1.2 FDOA

As stated by Loomis [1], the Doppler shift in signal frequency, $\Delta\phi$, due to the motion of a collector is given by

$$\Delta\phi = \frac{f_c}{c}\frac{\vec{v}\cdot\vec{r}}{|\vec{r}|}, \tag{3.7}$$

where f_c is the carrier frequency of the emitted signal, and c is the speed of light. In the terms of Figure 12, this means that the frequency shifts for each collected signal are

$$\Delta\phi_1 = \frac{f_{c_1}}{c}\frac{\vec{v_{c_1}}\cdot\vec{r_1}}{|\vec{r_1}|}, \text{ and} \tag{3.8}$$

$$\Delta\phi_2 = \frac{f_{c_2}}{c}\frac{\vec{v_{c_2}}\cdot\vec{r_2}}{|\vec{r_2}|}. \tag{3.9}$$

Since the two received signals are simply different copies of the same original signal, $f_{c_1} = f_{c_2} = f_c$. Thus, the frequencies of the signals received by the two collectors are

$$\phi_1 = f_c + \Delta\phi_1 = f_c + \frac{f_c}{c}\frac{\vec{v_{c_1}} \cdot \vec{r_1}}{|\vec{r_1}|} \text{ and} \tag{3.10}$$

$$\phi_2 = f_c + \Delta\phi_2 = f_c + \frac{f_c}{c}\frac{\vec{v_{c_2}} \cdot \vec{r_2}}{|\vec{r_2}|}. \tag{3.11}$$

The FDOA is simply the difference in these two received signal frequencies, and therefore

$$f = \phi_2 - \phi_1 = \frac{f_c}{c}\left[\frac{\vec{v_{c_2}} \cdot \vec{r_2}}{|\vec{r_2}|} - \frac{\vec{v_{c_1}} \cdot \vec{r_1}}{|\vec{r_1}|}\right]. \tag{3.12}$$

When $\vec{r_1}$ and $\vec{r_2}$ are allowed to be time-variant, and are defined by (3.5) and (3.6), (3.12) becomes

$$f = \frac{f_c}{c}\left[\frac{\vec{v_{c_2}} \cdot \vec{r_2}(t)}{|\vec{r_2}(t)|} - \frac{\vec{v_{c_1}} \cdot \vec{r_1}(t)}{|\vec{r_1}(t)|}\right]. \tag{3.13}$$

Thus we have accomplished our goal, as (3.13) now provides FDOA in terms of x, y, z and t.

3.1.3 Modified CAF

With (3.2) and (3.13) for TDOA and FDOA, respectively, it is now possible to determine a form of the CAF with position as its independent variables. Starting from the general form of the CAF as given in (2.1),

$$A(\tau, f) = \int_0^T s_1(t) s_2^*(t+\tau) e^{-j2\pi f t} dt, \tag{3.14}$$

where the signals $s_1(t)$ and $s_2(t)$ are of length T, thus bounding the integral.

To be practically implemented (3.14) must be converted to a discrete form to allow for digital signal analysis and computation. With f_s being the sampling frequency of the collected signals, and $T_s = {}^1/_{f_s}$ the sample duration, let

$$t = nT_s \tag{3.15}$$

and

$$f = \frac{kf_s}{N}, \tag{3.16}$$

where N is the number of collected samples for each signal, such that $T = NT_s$, and k is the digital FDOA. These substitutions lead to the discrete CAF,

$$A(\tau, k) = \sum_{n=0}^{N-1} s_1[n] s_2^* \left[n + \frac{\tau}{T_s} \right] e^{-j2\pi \frac{kn}{N}} \tag{3.17}$$

Substituting in (3.2) and (3.13), letting n_0 equal the first sample of the collected signals, and noting that $N = f_s T$, yields the modified CAF function,

$$A(x, y) = \sum_{n=n_0}^{n_0+f_sT-1} s_1[n] s_2^* \left[n + \text{ROUND}\left(\frac{|\vec{r}_2(nT_s)| - |\vec{r}_1(nT_s)|}{cT_s} \right) \right] e^{-j2\pi \frac{f_c n}{f_s c} \left[\frac{\vec{vc}_2 \cdot \vec{r}_2(nT_s)}{|\vec{r}_2(nT_s)|} - \frac{\vec{vc}_1 \cdot \vec{r}_1(nT_s)}{|\vec{r}_1(nT_s)|} \right]}. \tag{3.18}$$

In (3.18) the term $\left(\frac{|\vec{r}_2(t)| - |\vec{r}_1(t)|}{cT_s} \right)$ must be rounded to the nearest integer because it is in the digital domain and corresponds to an offset in the sample index, meaning that fractional offsets are not allowed.

3.2 Mapping Implementation

Equation (3.18) is implemented in two functions, LongIntCafMap.m and caf_map.m, which can be found in the Appendix. These functions will collectively be referred to as the Moss algorithm for the remainder of this work. The LongIntCafMap.m function defines the geometry of the collectors and emitter, the parameters of the signal, and the length and number of snapshots that will be collected. It then generates the "received" signals for each collector using the sig_gen.m function, written by Johnson in his thesis on CAF processing [11]. Although, in principle, the Moss algorithm is able to collect one long signal, this implementation continues to break it up into smaller snapshots. This is done for two reasons. First, we want to compare the Hartwell and Moss algorithms to determine any improvement made. Thus, we want to keep as many parameters as possible the same. If the Moss algorithm shows improvement using snapshots, then we can be reasonably confident that it would show even greater improvement with just one, long collection. Second, the trade-off for updating the system geometry every sample is very high computational complexity. This means that one long collection would take prohibitively long to simulate, and continuing to use the snapshots cuts down on this processing time greatly. After generating the "received" snapshots, the LongIntCafMap.m function passes them to the caf_map.m function.

The caf_map.m function accepts the collected signals and parameters for the size and resolution of the map. It then loops through every x and y position in the grid and calculates the CAF value at that position using (3.18). Unfortunately, this must be done in a loop rather than utilizing the FFT or cross-correlation methods because the parameters are changing in time, meaning that they must be updated every sample. As a result, this algorithm is unable to take advantage of the computational efficiency of those algorithms and must utilize brute force, as it were, by looping through every signal sample and computing each element individually. This causes a very high level of computational complexity, and thus, the algorithm is very slow. This is further discussed in the succeeding chapters. After calculating the CAF for each snapshot, caf_map.m returns it to LongIntCafMap.m, which coherently sums the values of all snapshots to yield the final CAFMAP result.

3.3 Key Differences

The next chapter will discuss the Moss algorithm in more detail and compare its results to those of the Hartwell algorithm. However, it is important to note the key differences between the two before proceeding.

1. Hartwell algorithm:
   (a) Snapshots are required.
   (b) Snapshots are summed non-coherently.
   (c) The CAF surface is binned into defined grid locations.
   (d) The FFT is used for computing CAFs; very efficient.
2. Moss algorithm:
   (a) Snapshots are not required; single, long duration samples may be used instead.
   (b) When snapshots are used, they are coherently summed to maintain phase information.
   (c) The CAF surface is computed directly in the xy domain.

(d) Each element of the CAF must be computed individually in loops; high computational cost.

In short, the Moss algorithm gains accuracy and resolution at the cost of computational complexity. The implications of this trade-off are fully examined in the next chapter.

CHAPTER 4: Results

The initial results show that the Moss algorithm does indeed locate RF emitters and maintains the CAFMAP's ability to simultaneously detect co-channel emitters as shown in Figure 13. This justifies the direct mapping solution used and opens the door for further work on the algorithm.

Having verified that the algorithm works as intended, the first task is to characterize it and check its performance compared to the original Hartwell algorithm. The computational complexity of the Moss algorithm turns out to have a large bearing on the ability to characterize the accuracy, so it is discussed first. This is followed by a side-by-side comparison of the results, obtained by using the two algorithms and a discussion on their differences.

4.1 Computational Complexity

Listing 4.1 shows the heart of the Moss algorithm as currently implemented in MATLAB. In order to analyze the computational complexity of the algorithm the methodology chosen is as follows. We assume that each operation requires one unit of time to perform, meaning that addition, subtraction, multiplication, division, assignments, etc. all take the same amount of time to perform. How true this assumption is will vary depending on the processor being used, but for an initial approximation, it is sufficient and simplifies the analysis. Additionally, the more complex operations, such as norms and vector multiplies, are broken up into their component basic operations. For example, the norm of a three element vector is $\text{norm}(\vec{x}) = \sqrt{x_1^2 + x_2^2 + x_3^2}$, so it is counted as three multiply, two add, and one root operations.

This done, the number of operations performed in each loop is counted and the result multiplied by the number of times that the loop is executed. The results of this are shown in Table 1. When the number of all operations are summed we arrive at # Ops $= 1 + N_x N_y (4 + 78N)$, where N_x and N_y are the number of gridpoints in the xy grid, and N is the number of collected samples for the signals. As an example, if we assume that the area of interest is an area 50 km wide and 20 km deep at a resolution of 10 meters, and the collection time is approximately 40 seconds at 100 kHz sampling rate, then the parameters

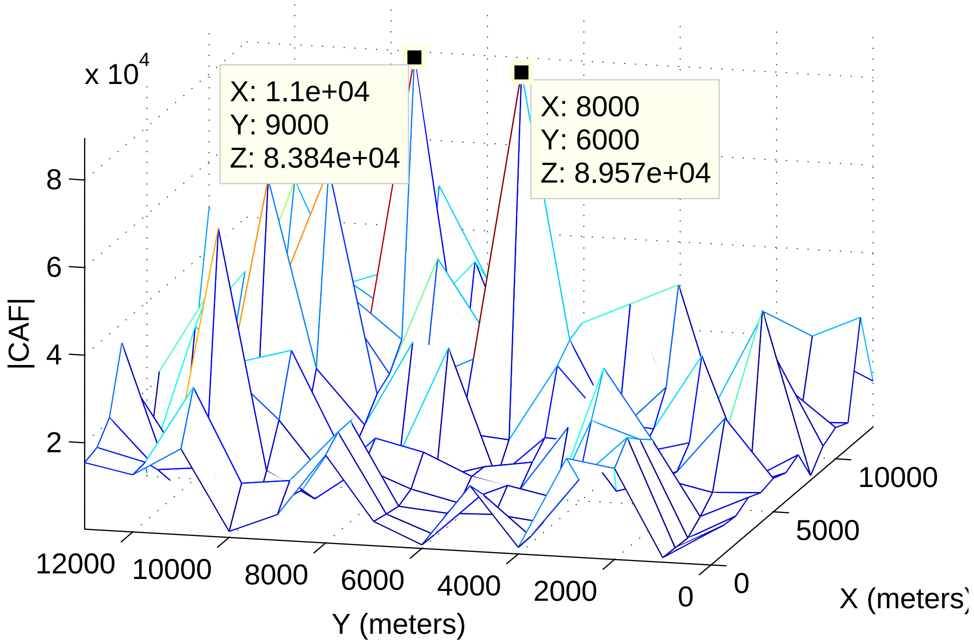


Figure 13. Moss algorithm with co-channel emitters at (10000,10000) and (6000,8000).

would be:

$$N_x = 5000,$$
$$N_y = 2000 \text{ and}$$
$$N = 2^{22}.$$

The total number of operations to be performed is $1 + N_x N_y (4 + 78N) = 3.27 \times 10^{15}$ and if it is assumed that one operation requires one nanosecond to perform then the total run time will be $(3.27 \times 10^{15})(1 \times 10^{-9}) = 3.27 \times 10^{6}$ seconds, or 37.86 days. This is obviously impractical, and time was not available to experimentally confirm this run-time. However, experiments were performed with either lower resolutions or shorter sample lengths and their run-times in MATLAB are shown in Figures 14 and 15. These figures show the run-time, in seconds, of each function called during the simulation. LongIntCafMap is the main function, and its total time is the total time taken for the entire simulation. The results, shown in these figures, confirm the complexity of the Moss algorithm and demonstrate its impracticality as currently implemented.

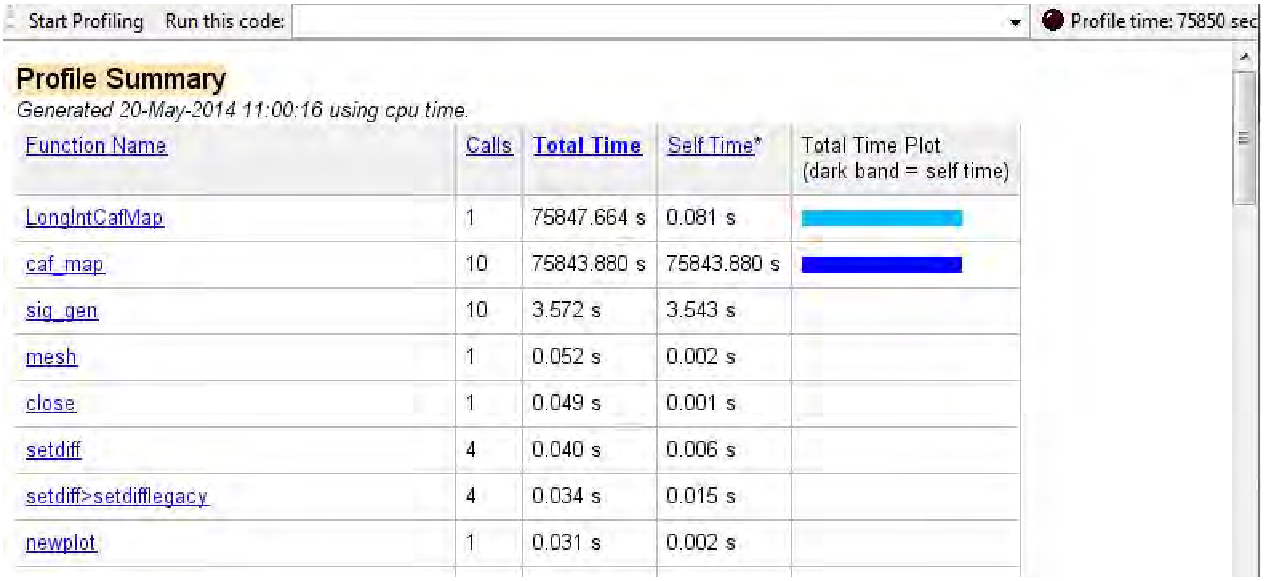
Listing 4.1. Moss algorithm in MATLAB code.

```matlab
for ii = 1:Nx
    for jj = 1:Ny
        m = zeros(1,N);
        exponent = zeros(1,N);
        Pe = [indexX(ii),indexY(jj),0];
        S2_conj = conj(S2);
        for nn = 1:length(S2)
            r1 = Pe - (Pc1+Vc1.*nn*Ts);
            r2 = Pe - (Pc2+Vc2.*nn*Ts);
            TauValue(nn) = round((norm(r2)-norm(r1))/c);
            if TauValue(nn) >= 0 && (nn+TauValue(nn)) < N
                m(nn) = S1(nn) * S2_conj(nn+TauValue(nn));
            elseif (nn-TauValue(nn)) < N
                m(nn) = S1(nn-TauValue(nn)) * S2_conj(nn);
            end
            FDOA(nn) = Fo/c*(Vc2*r2'/norm(r2)-Vc1*r1'/norm(r1));
            exponent(nn) = exp(-1j*2*pi*FDOA(nn)/Fs*(nn-1));
        end
        product = m.*exponent;
        CAF(ii,jj) = sum(product);
    end
end
```

Table 1. Number of operations performed.

Operation	# of times performed
Assign	$N_xN_y(10N+3)$
Add/Sub	N_xN_y29N
Multiply	$N_xN_y(26N+N+1)$
Divide	N_xN_y5N
Sqrt	N_xN_y4N
Compare	N_xN_y2N
Exponent	N_xN_yN

N_x = # of X grid positions mapped.
N_y = # of Y grid positions mapped.
N = # of signal samples taken.

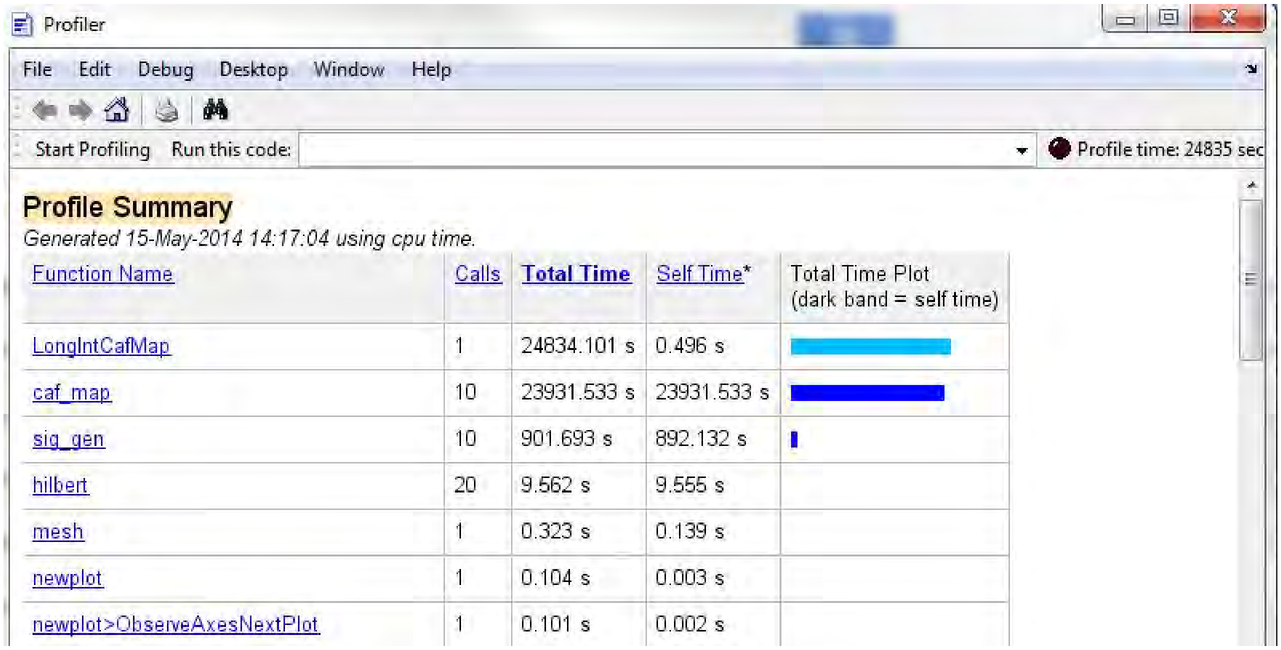


Profile Summary

Generated 20-May-2014 11:00:16 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
LongIntCafMap	1	75847.664 s	0.081 s	▬
caf_map	10	75843.880 s	75843.880 s	▬
sig_gen	10	3.572 s	3.543 s	
mesh	1	0.052 s	0.002 s	
close	1	0.049 s	0.001 s	
setdiff	4	0.040 s	0.006 s	
setdiff>setdifflegacy	4	0.034 s	0.015 s	
newplot	1	0.031 s	0.002 s	

Figure 14. $N = 2^{14}$, $dm = 100$ m, full map.



Profile Summary

Generated 15-May-2014 14:17:04 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
LongIntCafMap	1	24834.101 s	0.496 s	▬
caf_map	10	23931.533 s	23931.533 s	▬
sig_gen	10	901.693 s	892.132 s	▮
hilbert	20	9.562 s	9.555 s	
mesh	1	0.323 s	0.139 s	
newplot	1	0.104 s	0.003 s	
newplot>ObserveAxesNextPlot	1	0.101 s	0.002 s	

Figure 15. $N = 2^{22}$, $dm = 100$ m, small map.

4.2 Moss Algorithm versus Hartwell Algorithm

As discussed in the previous section, a full characterization of the Moss algorithm was unfeasible. In lieu of this, a selection of test cases were run with both the Moss and the Hartwell algorithms so that a rough comparison might be made and the general characteristics of the Moss algorithm determined. These tests were conducted with a lead-trail collector geometry, with the geometry parameters defined in Table 2. The transmitted signal is Binary Phase Shift Keying (BPSK) modulated with parameters as shown in Table 3. All tests are performed using both the re-created Hartwell algorithm and the Moss algorithm, with identical parameters, such that a direct comparison can be made.

Table 2. Test geometry parameters.

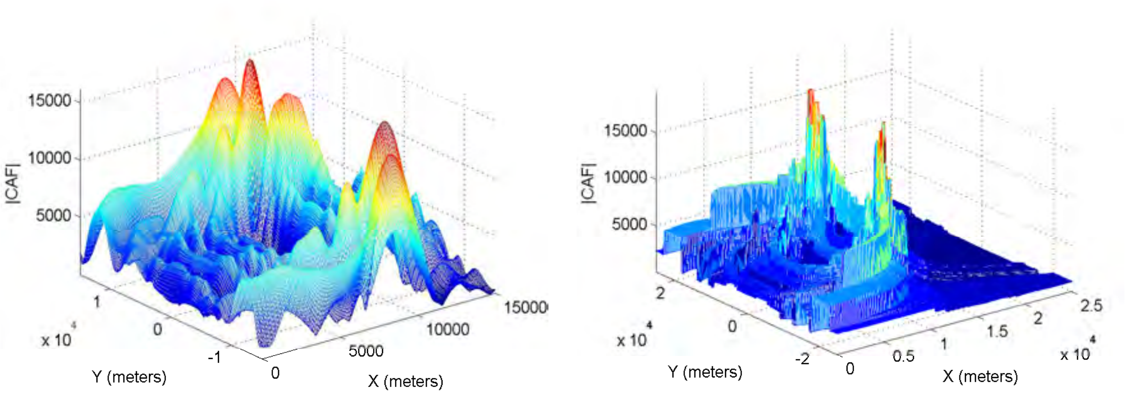
Parameter	Value
Collector 1 Position (P_{c_1})	[0, 0, 7500] m
Collector 1 Velocity (V_{c_1})	[100, 0, 0] m/s
Collector 2 Position (P_{c_2})	[2000, 0, 7500] m
Collector 2 Velocity (V_{c_2})	[100, 0, 0] m/s
Emitter Position (P_e)	[10000, 10000, 0] m
Emitter Velocity (V_e)	[0, 0, 0] m/s

Table 3. Transmitted signal parameters.

Parameter	Value
Carrier Frequency (f_c)	1000.025 MHz
Sampling Frequency (f_s)	100 kHz
Symbol Rate (R_{sym})	10000 symbol/sec
Signal-to-Noise Ratio (SNR) $\left(\frac{E_b}{N_0}\right)$	Variable

The first case chosen uses a signal length of $N = 2^{14}$ samples, a SNR of 10 dB, and a map grid size of $dm = 100$ meters. Furthermore 10 CAF snapshots were computed at 10 second intervals and then summed to provide the final surface. These parameters were chosen because the computations involved are short enough that a large map could be realistically generated by both algorithms and a large scale comparison made. Additionally, this case provides a reasonable baseline because the signal length is small enough that the Hartwell algorithm should have no difficulty and will perform perfectly fine. As shown in Figure

16, these expectations are realized. Both algorithms peak near the correct location of the emitter and show the expected ambiguity about the y axis due to the lead-trail collector configuration. Significant differences can be noted, however. The Moss algorithm provides a much smoother surface as each data point is computed independently and can take any value. The Hartwell algorithm, by contrast, maps the CAF surface to the geographic plane by placing the CAF magnitude values into xy bins. This binning behavior is not necessarily detrimental, but in some situations it could lead to losing some valuable information. Another observation of note from this test is that the smoothness of the Moss algorithm means that more of the CAF energy is located away from the peak. Since the peak is what shows the emitter's location, and is of interest, this dispersion could make geolocation more difficult with a greater spread in the Confidence Interval (CI).



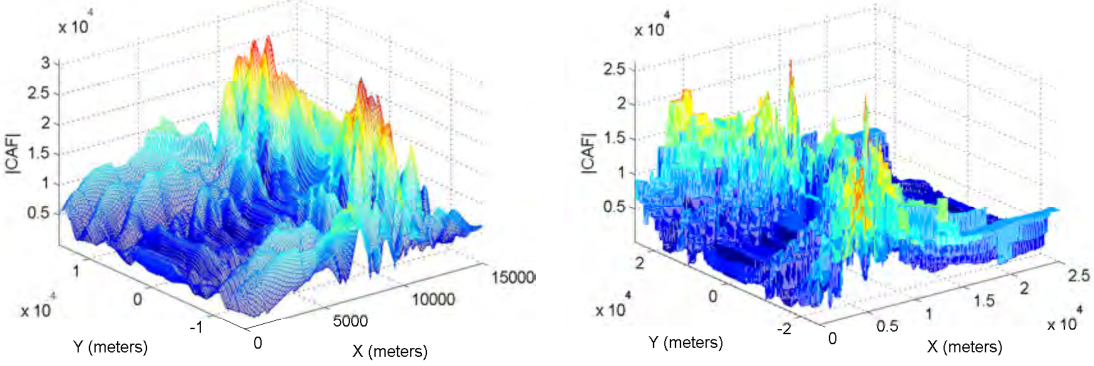
Moss algorithm

Hartwell algorithm.

Figure 16. $N = 2^{14}$, dm = 100, SNR = 10 dB, 10 Snapshots, Full Map.

The second test aims to see the effect of increasing the number of snapshots. These snapshots were Hartwell's method [5] of working around the signal length limitations of his algorithm by spacing out collection intervals to achieve isodop and isochrone rotation with the necessarily short collection times. In principle, this method of dividing the collection into snapshots is not necessary with the Moss algorithm; we should be able to just integrate straight through. However, in order to make one-to-one comparisons, and because of the computational complexity of long signals, the Moss algorithm will also use snapshots for these tests. Therefore, this test leaves all parameters equal to those of the first test, but changes the number of snapshots from 10 to 30.

As shown in Figure 17, the results are not entirely as expected. For the Hartwell algorithm, the peaks are indeed much narrower as desired, but the map also becomes much noisier. This is attributed two factors. First, Hartwell's algorithm only computes the magnitude of the CAF, which is then non-coherently summed, losing any phase information. This leads to less destructive interference occurring and the noise adding up with each snapshot. Second, the binning characteristic already discussed creates an 'on/off' behavior in CAF values at given *xy* coordinates. After the summation, this leads to some *xy* coordinates having very large values, while those right next to them may have very low values. The Moss algorithm does not show this characteristic. The surface is smooth, without any binning, and the surfaces are all summed coherently before taking the surface magnitude as the final step before plotting. This seems to reduce the clutter greatly and make additional snapshots more beneficial. This implies that the Moss algorithm would perform even better if we had the computational power to collect straight through, without snapshotting, but verifying this is left for future work.

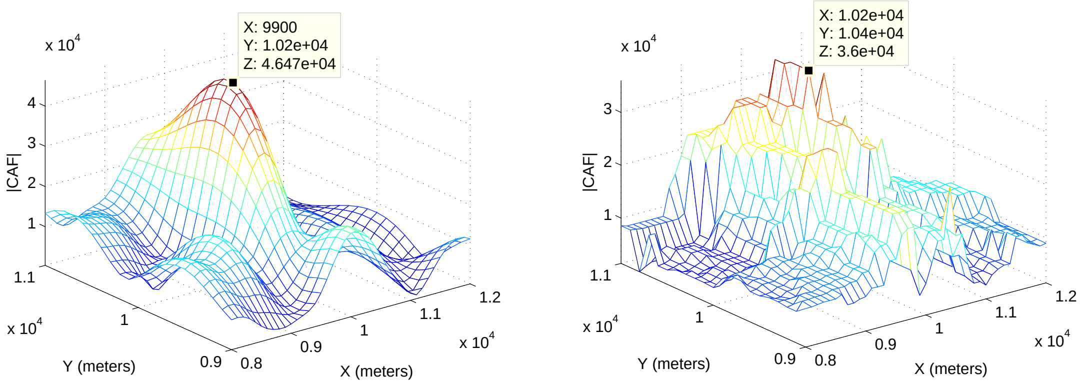


Moss algorithm

Hartwell algorithm

Figure 17. $N = 2^{14}$, dm = 100, SNR = 10 dB, 30 Snapshots, Full Map.

The third test investigates the effect of increasing the SNR to 30 dB, and also doubles the length of the signal. Additionally, this test reduces the map size so as to concentrate on just the area of interest. This is done both to reduce the computation time, and also because the first two tests have already shown the large scale behavior, which mostly follows expectations. The results are shown in Figure 18. The resolution used is still relatively large, and at this scale both algorithms perform adequately. However, it can be seen that smoothness of the Moss algorithm again makes it easier to find the true peak of the surface and determine an appropriate CI for the geolocation estimate. The Hartwell algorithm peaks nearly as close to the true emitter location as the Moss algorithm, but, there seems to be a greater variance in the peak and multiple locations are binned into the same value. This makes it difficult to determine the true peak.



Moss algorithm

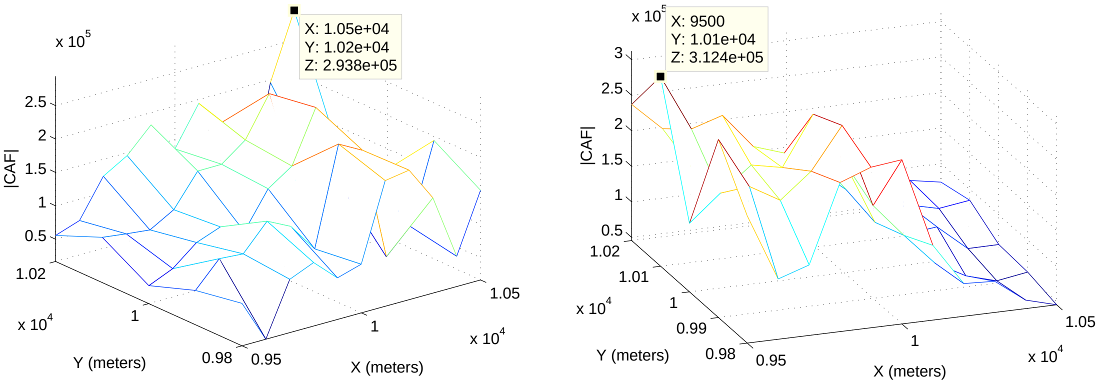
Hartwell algorithm

Figure 18. $N = 2^{15}$, dm = 100, SNR = 30 dB, 10 Snapshots.

The next test is an attempt to determine if the Moss algorithm is able to perform better than the Hartwell algorithm for long duration signals. Achieving this was the primary motivation for this work, and it is important see if it has been successful or not. As shown previously in Figure 11, the Hartwell algorithm begins to show significant smearing and loss of accuracy at signal lengths of 2^{21} so a length of 2^{22} was chosen for this test. With signals of this length the computation time becomes a serious problem so very small map sizes were used to make it practical.

The results of this test are shown in Figure 19. The first thing that we notice is that, this time, the Moss algorithm is just as rough as the Hartwell algorithm. This is simply the result of having a grid resolution on the same order as the grid size, the next test will correct this. Beyond the roughness, it can be seen is that both algorithms create a peak with approximately the same error. It appears that the Moss algorithm has a slightly lower (better) variance, but this is impossible to determine from just one test at this resolution, so no true conclusions can be drawn from this.

In order to draw a better conclusion, the test is re-done with a higher resolution of $dm =$ 10 meters. This increases the computation time by a factor of 10, so the grid size is halved



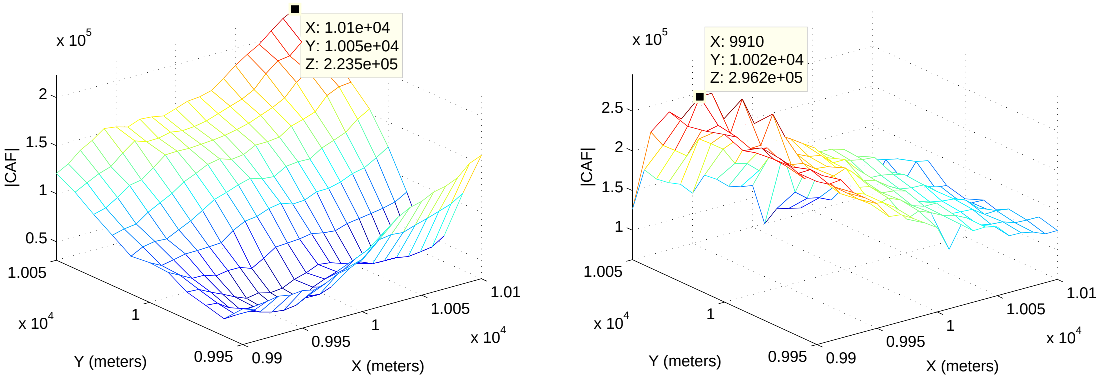
Moss algorithm

Hartwell algorithm

Figure 19. $N = 2^{22}$, dm = 100, SNR = 30 dB, 10 Snapshots.

to bring the increased computational complexity to a factor of 4. As shown in Figure 20, the results appear to be much better. The Hartwell algorithm's results appear smoothed out some and it arrives at a closer estimate for the emitter's position. However, this may be deceiving. The surface seems to be monotonically decreasing away from the peak, but this could merely be a result of using such a small grid size, and the true peak may be located off the grid. With this resolution and grid size it is impossible to tell if this is the true peak, or just a local peak. Furthermore, even if it is the true peak, the energy is quite dispersed over the grid, which will result in a higher variance in the maps energy. This means that the statistical geolocation accuracy will most likely be worse if multiple runs were performed with random noise.

The Moss algorithm, on the other hand, is able to show the true oscillating behavior of the CAF surface, which gives more confidence that we have found the true peak. The energy in the surface is also more concentrated near the peak—meaning that if more tests could be completed, and statistical analysis performed, the results are more likely to be consistent.



Moss algorithm

Hartwell algorithm

Figure 20. $N = 2^{22}$, dm = 10, SNR = 30 dB, 10 Snapshots.

4.3 Conclusion of Results

Without the ability to perform tests with both large-scale and high resolution maps, it is difficult to arrive at the true big-picture characteristics of the algorithm. Additionally, it is impossible to conclusively determine the accuracy and performance without the ability to do multiple runs of each test with random noise, and then perform statistical analysis to truly see what the mean and variance of the detection results are. However, short of these conclusive results, it appears that the Moss algorithm succeeds in geolocating RF signals with equivalent, or better, accuracy than the Hartwell algorithm, as shown in Figures 16-20. Furthermore, it is clear that the Moss algorithm provides a much better mapping resolution than the Hartwell algorithm. This suggests both better accuracy, and an improved ability to distinguish more subtle behaviors that would otherwise be smoothed out and missed. More work is necessary to draw convincing conclusions as to the utility of the Moss algorithm, but this chapter has shown that it merits further investigation. The final chapter will discuss the further work that needs to be accomplished in order to move this work forward, and the current thoughts on how to achieve it.

CHAPTER 5:
Future Work

This work has demonstrated that the Moss CAFMAP algorithm is capable of geolocating RF emitters. However, there is still a lot of work that should be done to truly complete this study, which will have to be left for a later work due to time constraints. This section will discuss the main aspects that require further investigation.

5.1 Reduce Complexity

In order to be practically useful, the algorithm must be able to run in less time, so that results can be obtained in near real-time as required by real world operations. The first, and simplest, method to improve the performance is to port the algorithm to a more suitable programming language, such as C or Fortran. These compiled languages are much better at handling large computations and loops, such as those present in the Moss algorithm. MATLAB was used in this work due to its ease of use, and the availability of previous code which could be used as a foundation. However, as an interpreted language, optimized for linear algebra, it does not perform well with the kinds of computations required in this algorithm. It is expected that such a port would provide substantial performance improvements, even if the algorithm itself were left in its current state.

5.1.1 Adaptive Resolution

Adopting an adaptive resolution regime is another method to reduce the complexity of the algorithm. Rather than try to geolocate the emitter in a single run it could be done in multiple stages, where each stage adapts the grid size, grid resolution, and signal length as appropriate until the desired accuracy is achieved. The initial stage could use a very short signal length and large grid with low resolution to find the approximate emitter location, and then the process can be repeated with a smaller grid, higher resolution, and longer signal length to iteratively arrive at a better solution. This concept is very similar to the coarse/fine method implemented by Johnson [11].

5.1.2 Parallelization

One great advantages of modern microprocessors is their multi-core architecture. This allows them to simultaneously perform multiple operations to reduce processing time. The current CAFMAP implementation does not make use of this capability, and is a factor that could be greatly improved. The algorithm should lend itself quite well to parallelization, as the same computations are performed with identical data many times with very well defined and independent changes. This has been shown true by Singh et al. as they successfully implemented the standard CAF in the 49-core Maestro processor, with significant reductions in processing time [12]. It should be fairly simple to break up the new Moss algorithm into parallel processing streams that can be recombined at the end. For example, the processing could be broken up either by grid locations, or by snapshots. Both ways would provide a set of long computations that are entirely independent and could be performed by separate processors.

5.1.3 Block Updates

In the current implementation of the Moss algorithm, the collector-emitter geometry is essentially being updated with every sample. While this reflects the true reality, it is useless in practice. At orbital velocities the collectors will be moving approximately 7.5 km/sec, and with a representative sampling frequency of 100 kHz, that means that the collectors will change position by ${}^{7.5\text{ km/sec}}/_{100\text{ kHz}} = 0.075$ m between each sample. So TDOA, for example, will only change by a maximum of $\frac{\Delta r}{c} \approx \frac{0.075}{3\times10^8} = 2.5\times10^{-10}$ sec between samples. This is a negligible difference, that is rounded out by the algorithm in its digital computation anyway.

Therefore, another approach that may possibly be used to reduce the complexity, would be to divide the collected signals into blocks of constant geometry, and only update collector positions when moving to a new block. By itself this would only simplify the algorithm marginally, but it may make it possible to use the FFT or correlation methods to calculate the CAF of each block. This would reduce complexity immensely, as those methods are far more efficient than brute force calculations. More work would need to be done to prove the feasibility of using the FFT or correlation functions however.

5.2 Experimentally Determine Statistical Results

Once the performance is improved, it is also important to gather statistical results on the accuracy and variations in the algorithm. It is impossible to completely characterize and determine the usefulness of this work without having these statistics experimentally determined. This is impossible to carry out with the current implementation, due to the computational complexity and long run times, but once those issues are resolved, the characterization of the algorithm needs to be completed.

5.3 Improve Geometry

Finally, this work has used a 2-Dimensional geometry for the emitter, with linear motion, in order to keep complexity at a minimum while performing the proof of concept for the work. However, this geometry is only realistic in limited situations, such as aircraft performing the collection over a small enough land area that can be accurately approximated as flat. For more general applicability, the algorithm must be fully extended to three dimensions, and for space applications, the collectors must follow an elliptical path for updating the geometry. Additionally, it is desirable to allow the emitter to be mobile, as well as the collectors, since this is often the case in real-world applications.

5.4 Conclusion

In conclusion, it is shown that the reformulated equations for TDOA and FDOA are accurate, and can be used in a modified CAFMAP algorithm to account for the motion of the collectors over time. The resulting Moss algorithm is able to geolocate RF emitters with accuracy similar to that of the original Hartwell algorithm for shorter collection times, and with apparent greater accuracy for long collection times. The improvement comes, however, with the cost of high computational complexity. This is significant enough that it must be mitigated before the new algorithm can be practically implemented, but the work required is merited. Additionally, the approach used here, of modifying the TDOA and FDOA equations to account for collector motion, should be generally applicable to many other methods of geolocation that estimate TDOA and FDOA as a fundamental step. This could lead to much broader use of these results, once the computational issues have been solved, and improve the geolocation accuracy of multiple methods in common use today.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: MATLAB Code

```
% This is an extension of Hartwell's CAF_MAP method and much of the
% initial code structure was taken from his work.
%
% This function defines all of the key parameters of the system,
% generates the signal vectors using Joe Johnson's sig_gen function,
% calls caf_map to generate the CAF_MAP in X and Y geographic
% coordinates, and then plots the combined result
%
% Usage: This is the initiating script, no inputs or
% outputs are required.
%
% Written by: LT Andrew Moss, Jan. 2014
% Last Modified: LT Andrew Moss December 2014

% clear all;close all;clc
close all

NUM_SNAPS = 30; % How many snapshots to take.
GAP = 10;       % Gap between snapshots in seconds.

Pc1 = [0,0,7.5e3];  % Collector 1 XYZ coordinates in meters.
Vc1 = [100,0,0];    % Collector 1 velocity, [Vx,Vy,Vz] in meters/sec.

Pc2 = [2e3,0,7.5e3];% Collector 2 XYZ coordinates in meters.
Vc2 = [100,0,0];    % Collector 2 velocity, [Vx,Vy,Vz] in meters/sec.

Pe = [10000,10000,0];% Emitter XYZ Position in meters.
Ve = [0,0,0];        % Emitter velocity in meters/sec.

f0 = 1000.025e6;    % Emitted carrier frequency in Hz.
fs = 100e3;         % Signal Sampling frequency in Hz.
Rsym = 10e3;        % Symbol rate for the generated BPSK signals.
N = 2^21;           % Length of collected signals, this also determines
                    % integration time.

Es_No1 = 30;        % Received SNR for collector 1.
```

```
Es_No2 = 30;        % Received SNR for collector 2.

dm = 100;           % Resolution of the generated XY map in meteres.

% This creates the XY grid.  Starts from Pe1 and makes a square to Pe2
% with a resolution of dm as specified above.  Units are meters.
Pe1 = [8.5e3,8.5e3];
Pe2 = [11.5e3,11.5e3];

t_snap = N/fs;      % The time covered by each snapshot in seconds.

indexX = Pe1(1):dm:Pe2(1);  % X grid points
indexY = Pe1(2):dm:Pe2(2);  % Y grid points
map_comb = zeros(length(indexX),length(indexY));    % initialize map.

% Break up the signals into snapshots and find the caf_map of each and
% sum the magnitudes
for ii = 1:1:NUM_SNAPS
    % Update Collector 1 position.
    Pc1_snap = Pc1 + ii*t_snap*Vc1 + (ii-1)*GAP*Vc1;
    % Update Collector 2 position.
    Pc2_snap = Pc2 + ii*t_snap*Vc2 + (ii-1)*GAP*Vc2;
    % Creates the signal vectors for the current snapshot.
    [Sa1, Sa2, S1_snap, S2_snap, Pcc1, Pcc2] = ...
        sig_gen(Pc1_snap,Vc1,Pc2_snap,Vc2,...
                Pe,Ve,f0,fs,Rsym,N,Es_No1,Es_No2);

    % Call the caf_map functions to generate the CAF surface
    % in the XY plane
    % for the current snapshot.
    [map,indexX,indexY]=caf_map(S1_snap,S2_snap,f0,...
                        fs,dm,Pe1,Pe2,Pc1_snap,Vc1,...
                        Pc2_snap,Vc2);

    % Add the current snapshot to the sum of the previous
    % snapshots.  All snapshots will have a peak at the actual
    % emitter location and will interfere both constructively
    % and destructively elsewhere.
    map_comb = map_comb + map;
```

```
end
% Plotting functions that may be used if found useful.
% [tdoa_grid, fdoa_grid, PtempX,PtempY] = ...
%   tdoa_fdoa_grid3d(Pc1_snap,Vc1,Pc2_snap,Vc2,Pe1,Pe2,f0,dm);
% Tau_Lo = round(min(min(tdoa_grid))*fs)-10;
% Tau_Hi = round(max(max(tdoa_grid))*fs)+10;
% Freq_Lo = (min(min(fdoa_grid))/fs)-.001;
% Freq_Hi = (max(max(fdoa_grid))/fs)+.001;
% [TDOA, FDOA, MaxAmb, Amb, TauValues,FreqValues] = ...
%                  caf_peak(S1_snap, S2_snap, Tau_Lo,...
%                  Tau_Hi, Freq_Lo, Freq_Hi, fs, 0);
% figure()
% f=mesh(TauValues/Fs,FreqValues*Fs,(abs(Amb)));
% title('CAF');
% set(f,'XData',TauValues);
% set(f,'YData',FreqValues);
% xlabel('TDOA');
% ylabel('FDOA');
% axis tight

savefile1 = 'map_comb.mat';
save(savefile1, 'map_comb');
% Plot mesh of the combined snapshots
figure()
hold on
f=mesh((abs(map_comb')));
set(f,'XData',indexX);
set(f,'YData',indexY);
title('')
xlabel('X (meters)')
ylabel('Y (meters)')
zlabel('|CAF|')
axis tight
grid on
set(gca,'FontSize',12)
set(findall(gcf,'type','text'),'FontSize',12)
```

```
function [CAF,indexX,indexY]=...
```

```
    caf_map(S1,S2,Fo,Fs,dm,Pe1,Pe2,Pc1,Vc1,Pc2,Vc2)
%**********************************************************************
%  [map,PtempX,PtempY]=caf_map(S1,S2,Fo,Fs,dm,Pe1,Pe2,Pc1,Vc1,Pc2,Vc2)
%  This function will calculate a CAF surface based upon input signals
%  S1 & S2 and map the caf to the 2 dimensional plane given by
%  Pe1 and Pe2.
%  Inputs:
%  S1      Signal from collector 1.
%  S2      Signal from collector 2 .
%  Fo      Carrier Frequency of signal.
%  Fs      Sampling Rate .
%  dm      Resolution in meters .
%  Pe1     Start of grid for Emitter's Position [X,Y] in meters.
%  Pe2     End of grid for Emitter's Position [X,Y] in meters.
%  Pc1     Position of collector 1 and beginning of snapshot [X,Y,Z].
%  Vc1     Velocity Vector of collector 1 [x,y,z].
%  Pc2     Position of collector 2 and beginning of snapshot [X,Y,Z].
%  Vc2     Velocity Vector of collector 2 [x,y,z].
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written By: LT Andrew Moss, April 2014
% Last Modified By: LT Andrew Moss, December 2014
% *********************************************************************
c = 2.997925e8; % Speed of light in m/s.
Ts = 1/Fs;      % Sampling period in sec.
N = length(S1);

% Builds the position vectors for the Emitter's Position
% Note this assumes a flat earth and the emitter is a 0m alt.
indexX = Pe1(1):dm:Pe2(1);  % X grid points in meters
indexY = Pe1(2):dm:Pe2(2);  % Y grid points in meters

% Get the dimensions of the map in order to go through each
% location.
Nx = length(indexX);
Ny = length(indexY);

% Initialize matrices for speed.
CAF = zeros(Nx,Ny);
TauValue = zeros(1,N);
```

```
FDOA = zeros(1,N);

% Start going through each map location and calculated the CAF value if
% that grid were the actual location of the emitter.
for ii = 1:Nx
    for jj = 1:Ny
        % Directly computing A(x,y) = sum{ s1(t)s2*(t,x,y)exp(j2pi t
        % f(t,x,y)) as derived in notes on 16OCT13
        m = zeros(1,N);
        exponent = zeros(1,N);
        Pe = [indexX(ii),indexY(jj),0];

        S2_conj = conj(S2);

        % Loop through each element of S2 for a given map location.
        for nn = 1:length(S2)
            % TauValue is calculated as a function of emitter position,
            % collector geometry and collector velocity.  Must be
            % rounded to a whole number.

            % Calculate relative position vectors for each collector and
            % the current map location.  Update each sample to account
            % for the position change of the collectors over time.
            r1 = Pe - (Pc1+Vc1.*nn*Ts);
            r2 = Pe - (Pc2+Vc2.*nn*Ts);
            % Find TDOA.
            TauValue(nn) = round((norm(r2)-norm(r1))/(c*Ts));

            % Perform the shift in the collected signals to account
            % for the TDOA and realign the signals.  Then calculate
            % the mixing function.
            if TauValue(nn) >= 0 && (nn+TauValue(nn)) < N
                m(nn) = S1(nn) * S2_conj(nn+TauValue(nn));
            elseif (nn-TauValue(nn)) < N
                m(nn) = S1(nn-TauValue(nn)) * S2_conj(nn);
            end

            % Calculate the FDOA from the relative position vectors.
            % This are done as vectors to avoid more loops, but as with
            % TDOA the value is changing in time.
            FDOA(nn) = Fo/c*(Vc2*r2'/norm(r2)-Vc1*r1'/norm(r1));

            % Calculate the exponential term of the CAF.
            exponent(nn) = exp(-1j*2*pi*FDOA(nn)/Fs*(nn-1));
        end
        % Finish calculating the CAF value for the current map position.
        product = m.*exponent;
        CAF(ii,jj) = sum(product);

% The following is debugging code, left in as it may be useful.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         fdoa_grid2(ii,jj) = mean(FDOA);
%         tdoa_grid2(ii,jj) = mean(TauValue);
%
%         T_vals = linspace(min(min(tdoa_grid2)),...
%                           max(max(tdoa_grid2)),length
%
%         %% Just Calc the CAF and then map it via Hartwell's method
%           % adds the 3rd dimensions at 0 meters in altitude
%         gridP = [indexX(ii),indexY(jj),0];
%
%         doppler1(ii,jj) = Fo/c * dot(Ve-Vc1, gridP-Pc1)...
%                            / norm(gridP - Pc1);
%         doppler2(ii,jj) = Fo/c * dot(Ve-Vc2, gridP-Pc2)...
%                            / norm(gridP - Pc2);
%
%         % Calculates the FDOA
%
%         fdoa_grid(ii,jj) = doppler1(ii,jj) - doppler2(ii,jj);
%
%
%         % Calculates the TDOA
%
%         tdoa_grid(ii,jj) = -(norm(gridP - Pc2) -...
%                            norm(gridP - Pc1)) / c;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
end
```

```
savefile1 = 'CAF.mat';
save(savefile1, 'CAF');
```

```
 function [Sa1, Sa2, S1, S2, Pcc1, Pcc2] = sig_gen(Pc1,Vc1,Pc2,Vc2,...
                                    Pe,Ve,f0,fs,Rsym,N,Es_No1,Es_No2)

% ***********************************************************************
% [Sa1, Sa2, S1, S2, Pc1, Pc2]] = sig_gen;
% SIG_GEN generates BPSK signal pairs based upon user-defined param-
%            eters and Cartesian emitter-collector geometries.  There are
%            no input arguments, since the function queries the user for
%            all required inputs.  The function returns four vectors:
%            Sa1, Sa2, S1 & S2.  These are the Analytic Signal represen-
%            tations of the two generated signals, and the Real represen-
%            tations of the two signals, respectively.
%
% Written by:  LCDR Joe J. Johnson, USN
% Last modified:  28 June 2005 By Glenn D. Hartwell
% Modified for 3D simulations and export collectors positions
% ***********************************************************************

clc;
% disp(' ');
% disp('All positions and velocites must be entered in vector format,');
% disp('e.g., [X Y Z] or [X, Y, Z] (including the brackets).');
% disp(' ');
%
% Pc1(1,:) = input...
%         ('Collector 1''s POSITION Vector at time 0 (in meters)?  ');
% Vc1 = input('Collector 1''s VELOCITY Vector (in m/s)?  '); disp(' ');
%
% Pc2(1,:) = input...
%         ('Collector 2''s POSITION Vector at time 0 (in meters)?  ');
% Vc2 = input('Collector 2''s VELOCITY Vector (in m/s)?  '); disp(' ');
%
% Pe(1,:) = input...
%              ('Emitter''s POSITION Vector at time 0 (in meters)?  ');
% Ve = input('Emitter''s VELOCITY Vector (in m/s)?  '); disp(' ');
%
% % f0 and fs are the same for BOTH collectors!
% f0 = input('Carrier Frequency (in Hz)?  ');
% fs = input('Sampling Frequency (in Hz)?  ');
Ts = 1/fs;      % Calculates Sample Period
%
% Rsym = input('Symbol Rate (in symbols/s)?  '); disp(' ');
Tsym = 1/Rsym;                      % Calculates Symbol Period
%
% N = input('How many samples?  '); disp(' ');
%
% Es_No1 = input('Desired Es/No at Collector 1 (in dB)?  ');
% Es_No1 = 10^(Es_No1/10);            % Converts from dB to ratio
%
% Es_No2 = input('Desired Es/No at Collector 2 (in dB)?  ');
% disp(' ');
Es_No2 = 10^(Es_No2/10);            % Converts from dB to ratio


Pc1 = [Pc1; zeros(N-1, 3)];     % Initializing all the matrices makes
Pe1 = zeros(N, 3);                  % later computations much faster.
Pc2 = [Pc2; zeros(N-1, 3)];
Pe2 = zeros(N, 3);
t1 = zeros(1, N);
t2 = zeros(1, N);
S1 = zeros(1, N);
S2 = zeros(1, N);

A = 1;                  % Amplitude of Signal
c = 2.997925e8; % Speed of light in m/s
Ps = (A^2)/2;       % Power of Signal

sigma1 = sqrt(Ps*Tsym/Es_No1);  % Calculate Noise Amplification fac-
sigma2 = sqrt(Ps*Tsym/Es_No2);  % tors using Es/No = Ps*Tsym/sigma^2

Noise1 = sigma1.*randn(N, 1);       % Random Noise values for Signal 1
Noise2 = sigma2.*randn(N, 1);       % Random Noise values for Signal 2
```

```
% Builds the position vectors for the two collectors
for index = 2 : N
   Pc1(index,:) = Pc1(index - 1,:) + Ts*Vc1;
   Pc2(index,:) = Pc2(index - 1,:) + Ts*Vc2;
end


% While loop determines first elements of Pe1 and t1.  t1(1) is the
% time AT THE EMITTER that produces the 1st sample received at
% collector 1!  Pe1(1,:) is the position of the emitter when it
% produces the 1st sample received by collector 1.

temp = inf;          % Ensures while loop executes at least once
t1(1) = 0;
tempPe = Pe(1,:);
while abs(temp - t1(1)) > 1/f0
   temp = t1(1);
   t1(1) = -norm(Pc1(1,:) - tempPe) / c;
   tempPe = Pe(1,:) + t1(1)*Ve;
end
Pe1(1,:) = tempPe;


% While loop determines first elements of Pe2 and t2.  t2(1) is the
% time AT THE EMITTER that produces the 1st sample received at
% collector 2!  Pe2(1,:) is the position of the emitter when it
% produces the 1st sample received by collector 2.

temp = inf;          % Ensures while loop executes at least once
t2(1) = 0;
tempPe = Pe(1,:);
while abs(temp - t2(1)) > 1/f0
   temp = t2(1);
   t2(1) = -norm(Pc2(1,:) - tempPe) / c;
   tempPe = Pe(1,:) + t2(1)*Ve;
end
Pe2(1,:) = tempPe;
```

```
% Platform positions at middle of snapshot
Pcc1=(Pc1(N/2,:));
Pcc2=(Pc2(N/2,:));
% Determines the earliest time at the emitter for this pair of signals.
StartPoint = min(t1(1), t2(1));


% Next 2 lines determine offsets needed for signals 1 & 2 to enter the
% phase vector (P).  This simply ensures proper line up so that bit
% changes occur at the right times.
SymbolIndex1 = 1 + floor(abs(t1(1) - t2(1))/Tsym) * (t1(1)>t2(1));
SymbolIndex2 = 1 + floor(abs(t1(1) - t2(1))/Tsym) * (t2(1)>t1(1));


for index = 2 : N                    % Builds the Pe1 and t1 vectors
   temp = inf;
   t1(index) = 0;

   % 1st guess is that emitter will advance exactly Ts seconds.
   tempPe = Pe1(1,:) + (t1(index -1) + Ts)*Ve;

   % While loop iteratively determines actual time & position for
   % emitter, based on instantaneous geometry.
   while abs(temp - t1(index)) > 1/f0
      temp = t1(index);
      t1(index) = (index - 1)*Ts - norm(Pc1(index,:) - tempPe) / c;

      % Due to negative times, must multiply Ve by ELAPSED time!
    tempPe = Pe1(1,:) + abs(t1(1)-t1(index))*Ve;
   end
   Pe1(index,:) = tempPe;
end


for index = 2 : N                    %Builds the Pe2 and t2 vectors
   temp = inf;
   t2(index) = 0;

   % 1st guess is that emitter will advance exactly Ts seconds.
   tempPe = Pe2(1,:) + (t2(index -1) + Ts)*Ve;

    % While loop iteratively determines actual time & position for
   % emitter, based on instantaneous geometry.
   while abs(temp - t2(index)) > 1/f0
      temp = t2(index);
    t2(index) = (index - 1)*Ts - norm(Pc2(index,:) - tempPe) / c;

      % Due to negative times, must multiply Ve by ELAPSED time!
      tempPe = Pe2(1,:) + abs(t2(1)-t2(index))*Ve;
   end
   Pe2(index,:) = tempPe;
end


% Could change this seed to whatever you want; or could have user
% define it as an input.  This just ensures, for simulation purposes
% that every time the program is run, the BPSK signals created will
% have the same random set of data bits.
rand('seed',5);

% Create enough random #'s to figure phase shift (data bits)
r = rand(N,1);
P = (r > 0.5)*0 + (r <= 0.5)*1;     % Since BPSK, random # determines
                                                % if phase is 0 or pi



% Building Xmitted Signal #1 vector...  These represent the pieces of
% the signal that were transmitted by the emitter to arrive at
% Collector 1 at its sample intervals.

S1(1) = A*cos(2*pi*f0*t1(1) + P(SymbolIndex1)*pi) + Noise1(1);

% The if statement inside the loop changes the data bit if the time
% has advanced into the next symbol period.
for index = 2 : N
   if t1(index) - StartPoint >= (SymbolIndex1) * Tsym
      SymbolIndex1 = SymbolIndex1 + 1;
   end
   S1(index) = A*cos(2*pi*f0*t1(index) + P(SymbolIndex1)*pi) + ...
                Noise1(index);
end

Sa1 = hilbert(S1);  % Calculates the ANALYTIC SIGNAL of S1.  To
                            % compute the COMPLEX ENVELOPE, multiply Sa1
                            % by .*exp(-j*2*pi*f0.*t1);


% Building Xmitted Signal #2 vector...  These represent the pieces of
% the signal that were transmitted by the emitter to arrive at
% Collector 2 at its sample intervals.

S2(1) = A*cos(2*pi*f0*t2(1) + P(SymbolIndex2)*pi) + Noise2(1);

% The if statement inside the loop changes the data bit if the time
% has advanced into the next symbol period.
for index = 2 : N
   if t2(index) - StartPoint >= (SymbolIndex2) * Tsym
      SymbolIndex2 = SymbolIndex2 + 1;
   end
   S2(index) = A*cos(2*pi*f0*t2(index) + P(SymbolIndex2)*pi) + ...
                Noise2(index);
end

Sa2 = hilbert(S2);  % Calculates the ANALYTIC SIGNAL of S2.  To
                            % compute the COMPLEX ENVELOPE, multiply Sa2
                            % by .*exp(-j*2*pi*f0.*t2);


% This function call simply calculates and displays the expected TDOAs
% and FDOAs at the Beginning and End of the collection time.

% tdoa_fdoa(f0,Pe1(1,:),Pe1(N,:),Pe2(1,:),Pe2(N,:),Ve,Pc1(1,:),...
%          Pc1(N,:),Vc1,Pc2(1,:),Pc2(N,:),Vc
```

List of References

[1] H. H. Loomis, “Geolocation of electromagnetic emitters,” Naval Postgraduate School, Monterey, CA, Tech. Rep. NPS-EC-00-003, October 2007.

[2] D. M. G. Price, “Mathematics of geolocation,” 2002, unpublished.

[3] P. D. Groves, *Principles of GNSS, Intertial, and Multisensor Integrated Navigation Systems*. Norwood, MA: Artech House, 2013.

[4] A. Buczek, “Notes and private conversations at Naval Research Laboratory,” Dec. 2002, unpublished.

[5] G. D. Hartwell, “Improved geo-spatial resolution using a modified approach to the complex ambiguity function (CAF),” master’s thesis, Dept. Elec. Eng., Naval Postgraduate School, September 2005.

[6] S. Stein, “Algorithms for ambiguity function processing,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 588–599, 1981.

[7] D. J. Torrieri, “Statistical theory of passive location systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, pp. 183–198, 1984.

[8] D. C. Shin, “Complex ambiguity functions using nonstationary higher order cumulant estimates,” *IEEE Transactions on Singal Processing*, vol. 43, pp. 2649–2664, November 1995.

[9] A. Ramirez, I. Rivera, and D. Rodriguez, “SAR image processing algorithms based on the ambiguity function,” in *Circuits and Systems, 2005. 48th Midwest Symposium on*, Aug 2005, pp. 1430–1433 Vol. 2.

[10] M. Skolnik, *Radar Handbook*. Boston, MA: McGraw-Hill, Inc., 1990.

[11] J. J. Johnson, “Implementing the cross ambiguity function and generating geometry-specific signals,” master’s thesis, Dept. Elec. Eng., Naval Postgraduate School, September 2001.

[12] K. Singh *et al.*, “FFTW and complex ambiguity function performance on the maestro processor,” in *Aerospace Conference, 2011 IEEE*, March 2011, pp. 1–8.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California